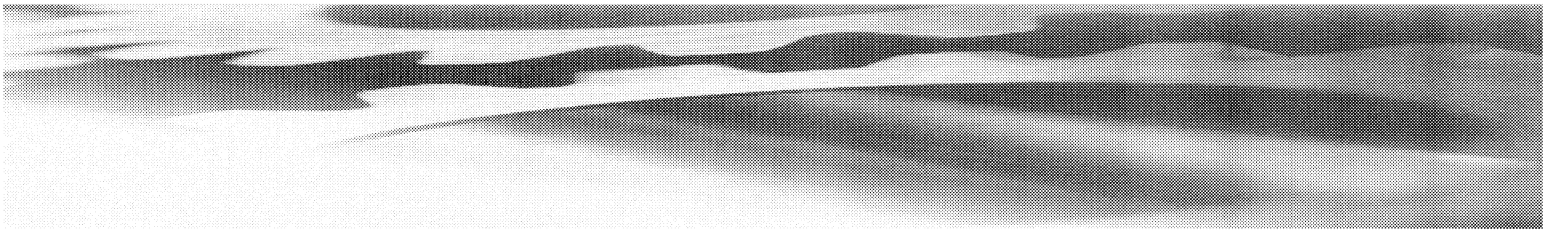


Advanced Lab Workbook

Motion Control in the ControlLogix 5550 System

Rockwell Automation

Welcome

Welcome to Allen-Bradley Hands On Motion!

The goal of Hands On Motion is to do just that — get your hands on Motion Control products quickly, so you can learn by doing.

During this workshop we will discuss the products, answer your questions, and spend most of the time in lab.

During Hands On Motion, feel free to ask questions. Let's make this time as valuable as it can be for you.

When you have finished this workshop, please complete an evaluation form. Tell us what more we can do to educate you on our products.

We would be happy to discuss how our Motion Control Products will fit your applications.

Thank you for attending this workshop. Ready to get started?

Lab Checklist

You will need the following items for this lab:

- ControlLogix demo unit with Logix5550 controller
- RSLogix 5000 and RSLinx software loaded on PC
- ControlLogix setup files loaded on PC
- ControlLogix System brochure (Publication 1756-1.3)

Ground Rules

If you are sharing a demo unit, make sure each of you has equal hands-on time.

Don't jump ahead in the workbooks; stay with the class. If you want some more challenging exercises after completing those in the workbook, ask the instructor.

If you have a question, ask it. This is your time to learn.

Before You Begin

- What other Hands On Motion workshops have you completed?
- What have you heard about the ControlLogix system?
- What questions do you have as we begin this lab?

Let's get started!

Contents

Up To Speed ..... Page 5

Understand a simple Motion Control System..... Page 6

Review the Control Logix System..... Page 7

Review the Logix 5550 in the Control Logix System..... Page 9

Review RSLogix 5000 in the Control System ..... Page 11

Review Integrated Motion in the Control Logix System ..... Page 15

Understand Motion Enhancements of RSLogix 5000, V2.25..... Page 19

Check Your Configuration (Demo Setup) ..... Page 25

Lab 1, Develop an application to create virtual_proc_1 ..... Page 26

Lab 2, Create a Consumed Axis associated with Processor slot 1..... Page 32

Lab 3, Develop an application to Gear Encoder_y to Motor_x then Move while gearing is enabled..... Page 36

Lab 4, Perform Position Cam Moves, create Cam profiles using graphical CAM Editor ..... Page 42

Lab 5, Perform Time Cam Moves, create Cam profiles using the MCCP instruction ..... Page 53

Appendix A, How To Configure an ETHERNET Card ..... Page 60

Appendix B, Create a Virtual Axis..... Page 62

Appendix C, Create a Consumed Axis ..... Page 68

Up To Speed

On Logix5550 Controller

Objectives

Understand a simple Motion Control System

Review the Control Logix System

Review the Logix 5550 in the Control Logix System

Review RSLogix 5000 in the Control System

Review Integrated Motion in the Control Logix System

Understand Motion Enhancements of RSLogix 5000, V2.25

Reference material

ControlLogix System brochure (Publication 1756-1.3)

What is Motion Control?

Motion Control is the precise control of mechanical motion. It means managing the speed, position, acceleration, deceleration and torque of an industrial system in a production line.

Allen-Bradley is a leader in motion control because they offer:

- Total automation solutions
- Easy-to-use operator interfaces
- An outstanding organization for sales, service and parts, and
- The ease of doing business with a company that is known world-wide for quality, technology expertise and customer satisfaction. Any closed motion control system consists of at least five key components:
- The load — what is being moved.
- The motor — the power behind the move; the force that causes the motion.
- Feedback device — resolver, encoder, or other device that communicates what is happening.
- The drive — the propelling mechanism in a machine; Takes the signal to the motor to move the load.
- The motion controller — Takes the signal to the drive. Use of software and communications that are used to build, monitor and troubleshoot applications.

The Key Reasons for Motion Control

Let's imagine that you have a load, and you want to move it from point A to point B. You connect a drive — perhaps an arm that swings from point A to point B. You have a motor to power the move. And you have an encoder to give feedback about what's happening.

But when should the move begin? When should it stop? Under what conditions? How far should it move? When will it know that it is in the correct position?

Motion control is the "glue" that connects all the other parts of an assembly line. It manages the motion!

Well-designed motion control moves things quickly and precisely to improve work processes, increase productivity and maximize business profit.

The ControlLogix System

ControlLogix is a revolutionary change in industrial automation.

This new **control architecture** from Allen-Bradley integrates communications and a programmable sequential controller in one state-of-the-art system. Because ControlLogix combines multiple disciplines into a single platform, you can reduce controllers and programming packages into a **tightly integrated system**.

And no other platform offers the **connectivity to legacy systems** with the complete ability to move forward!

Key Features and Benefits

Increased distribution of control

- Multiple controllers over ControlNet or in a single chassis
- Flex I/O over various networks to put I/O with the devices

Scaleable control system options

- Designed for multiple controller support
- Future form factors
- Complete modularity of communication, controllers and memory
- System monitors modules for health, correct placement and configuration control

Increased **use of commercial technology**

- Commercial technology used where appropriate
- Windows NT for programming and standard Microsoft data links
- Ethernet as the information connection

Open communication for system-wide linking

- Totally modular communication for flexibility
- State-of-the-art producer-consumer model for networks and backplane

Reduction of controllers and programming packages

- Integration of multiple disciplines into a single platform
- Multiple control data bases become a single data base

Higher functionality in a smaller, lower priced form factor

Backward and forward **migration**

Integrated Disciplines

How many different intelligent boxes do you use, each with its own operating characteristics and programming software? What about different databases used on the plant floor?

Wouldn't you like to reduce the confusion with an open, integrated system?

ControlLogix uses a common base of I/O, communication interfaces, a single controller and programming software to address multiple control applications.

- Reduce the number of boxes.
- Simplify the control and programming.
- Decrease learning curve and maintenance issues.

Advanced Networking Capabilities

A ControlLogix system looks like a programmable controller, but is much more. Unlike conventional programmable controllers, it was designed with **communications at its core**, reducing system bottlenecks.

ControlLogix breaks the PLC mold by using the same contemporary model found in ControlNet and DeviceNet (Producer-Consumer). By buying only what you need, you can design a control system to match your applications. Easily add application modules and networks in the future as needed.

- ControlBus backplane
- Producer-consumer model
- Network choice

What is ControlBus?

ControlBus eliminates the controller as the center of the control universe. Any device in the backplane can broadcast and communicate with any other device without controller intervention. This increases system bandwidth and performance, and gives you greater flexibility in building a control system.

Traditional control systems use a monolithic master/slave address bus. This means that a single device rules the bus and controls all traffic. If the controller or bus-arbiter dies, then you have a single point of failure.

In contrast, ControlBus provides a multi-master producer/consumer model for communication. This requires less programming and increases system throughput.

The Logix5550 Controller

The Logix5000 series of controllers, together with RSLogix5000 software, are **power packed into an easy-to-use environment**.

The Logix5550, the first in the series, operates in the ControlLogix architecture using the ControlBus backplane to **rapidly communicate with other controllers, networks and I/O modules**.

Greater functionality, faster performance, smaller size, lower cost, and scaleable in all dimensions.

- **Outperforms** other controllers in its class.
- Requires 20% to 50% **less panel space** than comparable controllers.
- Highest functionality at a system price between the PLC5 and SLC500.

Multiple controllers in one small package.

- Highly integrated motion capability saves you money by eliminating separate motion controllers and software.
- Enhancements for process and drive systems.

Fully scaleable solution allows you to purchase exactly what you need.

- Memory expandable from small to large applications.
- **Modular networks** provide the right communications mix for your application: DH+, RIO, ControlNet, DeviceNet, EtherNet.
- I/O capacity sufficient to meet large applications.
- Increase performance from distributed processors.

Includes an **RS232 interface port** that acts as your portal into the system

- Eliminate the need for computer cards.
- Provide **remote diagnostic** support.
- Support DF1 protocol for programming terminals and other intelligent devices to access data and upload/download programs.

Distributed control solution.

- Multiple controllers in a single chassis perform load-sharing to increase application performance.
- CPUs distributed across ControlNet while sharing I/O.
- Processor -to-processor communications without writing application codes.

*The **powerful Logix5550 proves itself** in application after application. Why? Because it takes a distributed approach to control that unburdens it from the overhead associated with communication and I/O tasks. The result: a controller focused on execution of its rich instruction set.*

Logix5550 I/O Capacity

In most PLCs today, a portion of memory is reserved to store the I/O values in the system. This places an artificial limit on the amount of I/O that a PLC can serve. In contrast, the Logix5550 allocates memory as you assign modules so **memory is no longer a limiting factor**.

The advantage of assigning memory dynamically is that discrete and analog modules communicate using the same approach, eliminating the need for block transfers to ControlLogix I/O.

Each module is assigned a data structure with real names, so you won't have to dig through a manual to figure out what the third word of an analog module is used for. Example: Local:1:I.Ch0Data

With a 250-connection limit for the CPU, you can talk to a maximum of 128-thousand discrete points or 4000 (250 x 16 analog points per card) analog channels. Messaging also uses some of these connections, but they are opened when needed and closed when the message is completed. This allows a small pool of connections to serve a large number of messages.

The Logix5550 Instruction Set

How similar is the Logix5550 instruction set to PLC5 and SLC500? We've brought the best of the PLC/SLC instruction set into Logix5550 to provide a migration path for PLC/SLC users. In fact, you'll have a PLC/SLC translation tool built into RSLogix 5000 to bring logic from these controllers forward. You can also cut/copy/paste logic between RSLogix 5 and RSLogix 500.

A new feature is the ability to do 32bit math natively, versus 16 bit in previous products. We've optimized the instruction set to work with 32bit data.

In addition to the PLC/SLC instructions, we've added a suite of Motion Control ladder diagram instructions so you can write one program for both PLC and Motion Control applications.

- **Same look, feel** and execution as existing products
- Reduced learning curve for new controller
- **Powerful** file and diagnostic instructions
- PID and other process instructions
- 30 Motion Instruction based on the Allen-Bradley GML environment
- **One program environment** for motion, sequential and process control

RSLogix Single Programming Interface

RSLogix 5000 is Windows NT-based programming software for the Logix5000 family of controllers. Along with other RSLogix software programs, we've provided you with a single Windows programming interface for all Allen-Bradley controllers:

RSLogix 500 for…	SLC 500
RSLogix 5 for…	PLC5
RSLogix 5000 for…	Logix 5000

- These applications work in the same box without re-boot.
- RSLinx and RSNetWorx provide common communications driver
- Cut/copy/paste between the controller programs
- 32-bit Windows allows you to take advantage of commercially available technology via OLE2, DDE, ActiveX, etc.
- Easily integrate additional capabilities and products into your control solution
- RSLinx allows your personal computer to communicate to any controller either in the chassis or distributed on a network.

RSLogix 5000 is the only programming software needed to fully configure and program an integrated Logix5550 motion solution. Software motion features automate what was often a manual motion control development process.

Quiz Yourself

Take a few minutes to answer the following questions.

About ControlLogix

1. True or False: A key benefit of the ControlLogix system is that you can reduce controller and programming packages because it combines multiple disciplines into a single platform.

2. True or False: ControlLogix platform offers connectivity to legacy systems.

3. What networking model is used in ControlLogix, as well as ControlNet and DeviceNet?

   a. Master-Slave

   b. Producer-Consumer

About Logix5550 Controller

4. True or false: ControlLogix integrates to motion and sequential control within a single controller by embedding motion control functions in RSLogix 5000 programming software and the Logix5550 controller.

5. Which of the following communications can be used with Logix5550 controller?

a. ControlNet and DeviceNet only

b. DH+, RIO, ControlNet, DeviceNet, EtherNet

c. ControlNet, DeviceNet and Ethernet only

d. Ethernet only

About RSLogix5000 and RSLinx Software

6. True or false: You can freely cut/copy/paste among the controller programs, including SLC 500, PLC 5, Logix 5000, RSLogix 5, RSLogix 500 and RSLogix 5000.

7. True or false: RSLinx allows your personal computer to communicate to any controller either in the chassis or distributed on a network.

8. True or false: RSLogix 5000 has a built-in PLC/SLC translation tool to bring logic from PLC and SLC controllers forward.

9. Correctly sequence the following steps in creating motion control via ControlLogix:

a. Step _____: Place 2-axis servo module(s) in ControlLogix chassis and wire to drives.

b. Step _____: Use RSLogix 5000 software to develop ladder logic program using any of 30 motion instructions.

c. Step _____: Use RSLogix 5000 software to name and configure each motion axis (up to 32 per Logix5550 controller).

d. Step _____: Download application program to Logix5550 controller and run.

10. True or false: RSLogix 5000 is the only programming software needed to fully configure and program an integrated Logix5550 motion solution.

Product Review Answer Key

About ControlLogix

1. True.

2. True.

3. Producer-Consumer.

About Logix5550 Controller

4. True.

5. DH+, RIO, ControlNet, DeviceNet, EtherNet

6. True.

7. True.

8. True.

9. A (step 1), b (step 3), c (step 2), d (step 4)

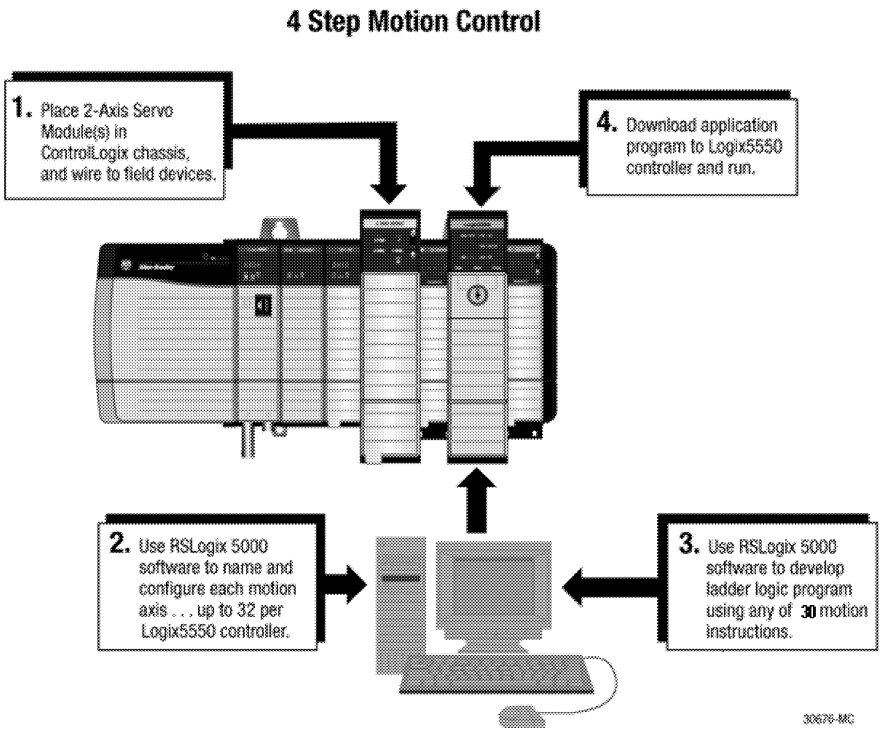
10. True.

Integrated Motion Control

The ControlLogix system takes a new approach to integrating motion and sequential control. ControlLogix performs these functions within a single controller, while other control systems require multiple controllers and programming packages.

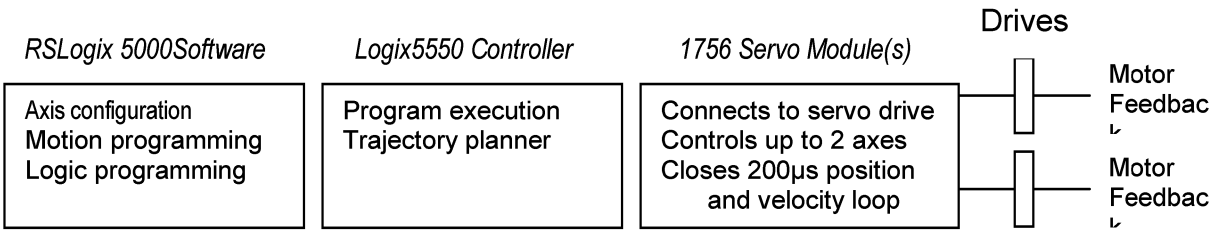
The ControlLogix system achieves this by embedding motion control functions in RSLogix 5000 programming software and the Logix5550 controller. The ControlLogix approach results in faster application development and integration, higher system performance and greater ease of start-up and maintenance.

An integral part of the Logix5550 controller's instruction set is a set of **30 motion commands**, including: homing, jogging, point-to-point positioning, gearing, high-speed position registration, and auto-tuning.


4 Step Motion Control
1. Place 2-Axis Servo Module(s) in ControlLogix chassis, and wire to field devices.
4. Download application program to Logix5550 controller and run.
2. Use RSLogix 5000 software to name and configure each motion axis . . . up to 32 per Logix5550 controller.
3. Use RSLogix 5000 software to develop ladder logic program using any of 30 motion instructions.
30676-MC

The Logix5550 controller, 1756-M02AE servo module and RSLogix5000 programming software provide integration motion control support for ControlLogix system.

It's important to understand the role each of these components plays in a motion control solution for a ControlLogix system.



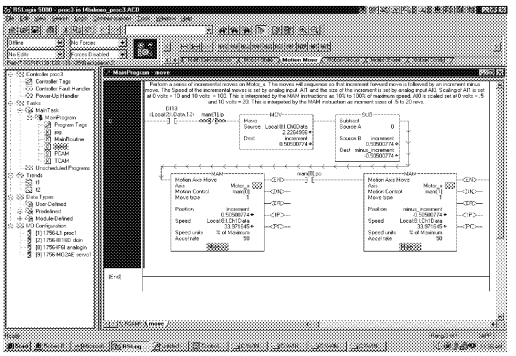
RSLogix Software

Use RSLogix Software for complete axis configuration and motion programming support. Use ladder logic to create an application program.

The RSLogix 5000 programming software provides complete programming and commissioning support for the ControlLogix system. RSLogix 5000 is the only programming software needed to fully configure and program the ControlLogix motion control systems.

RSLogix 5000 software provides the following motion support:

- Wizards for servo axis configuration including drive hookup diagnostics and auto-tuning
- Ladder-based application programming including 30 types of motion commands
- Insert motion instruction(s) as needed and define motion parameters
- Download to the Logix 5550 controller and run
- Support for on-line monitoring and editing



Logix5550 Controller

The Logix5550 controller executes the ladder motion commands. It also creates position and velocity profile information that is sent to the servo module(s).

The Logix5550 controller is the main component in the ControlLogix system. It supports sequential and motion functions, and it performs all of the motion command execution and motion trajectory planner functions. You can use one or more Logix5550 controllers in each chassis, and each controller can control up to 16 motion modules.

The Logic5550 controller provides the following motion support:

- 30 motion instructions
- 4 Axis types supported: Servo, Position Only, Virtual and Consumed
- A high-speed motion task, which manages motion functions and generates move profiles. Performance is roughly a linear function of the number of axis.
- The ability to control up to 16 Analog/Encoder Servo modules for a total of 32 axes

1756 Servo Module

The 1756-M02AE Servo Module connects to a servo drive and closes a high-speed position and velocity loop.

Each Logix5550 controller can support up to 16 1756-M02AE servo modules. Each 1756-M02AE module can control up to two axes.

The Analog/Encoder Servo Module provides an analog/quadrature encoder servo drive interface. The servo module receives configuration and move information from the Logix5550 controller and manages motor position and velocity.

The servo module supports the following.

Connection capability for up to two drives:

- 16-bit analog outputs
- Quadrature encoder inputs
- Home limit switch inputs
- Drive fault inputs
- Drive enable outputs
- 5V or 24V registration inputs
- 200 µsec position and velocity loop updates

The 1756 Module can be used with any drive that supports analog voltage input and a quadrature encoder ouptut including:

- 1394-SJTxx-A…1394 Multi-axis servo drive
- 1398-DDM-xxx….Ultra 100 & 200 single axis servo drive
- 1336 Force
- 1336 Impact

Pre-terminated cables are also available for use with the 1394 and 1398 servo drive.

Motion CAM Instructions

There are three new motion instructions in the new release of RSLogix 5000, V2.25:

- Motion Calculate Cam Profile (MCCP)

  Use the MCCP command anytime you want to enter data points or change data points using RSLogix 5000 for either a MAPC or MATC instruction

- Motion Axis Position Cam (MAPC)

  Use the MAPC command anytime a slave axis must follow a master axis via a complex profile.

- Motion Axis Time Cam (MATC)

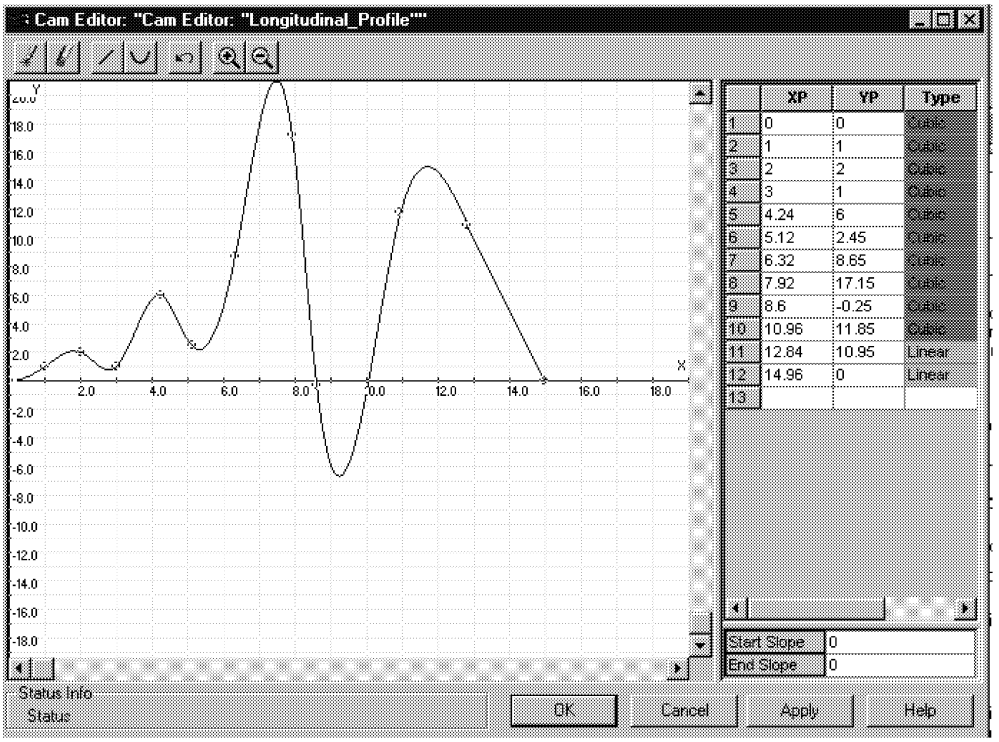
  Use the MATC command anytime a complex motion profile is required.

These Cam instructions utilize either Linear and Cubic profile segment for whatever best fits the application.

All profiles can be created and downloaded with the application program or created dynamically in the application program.

RSLogix 5000 CAM EDITOR

A graphical CAM profile editor is built into the RSLogix 5000 software.



You can use this CAM profile editor to create profiles as part of the application program development in several ways:

- By editing in the graph and grid view

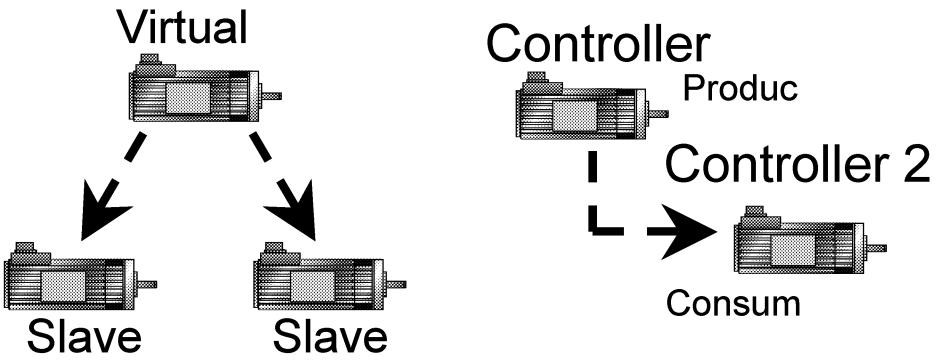
- Using linear and cubic spline options for each segment

- Cut and paste data from spreadsheets (Excel for instance)

You can also monitor profiles created dynamically during the routine and all profiles can be dynamically created or modified in the application program using the MCCP motion instruction.

Motion Virtual/Consumed Axis

Two new types of motion axes have been added to the Control Logix System: Virtual and Consumed



Virtual Axis

A Virtual Axis has all the attributes of a servo axis except there is no physical ouput. This is included for all motion move support including gear and CAM.

A Motion Module is not required to create a Virtual Axis.

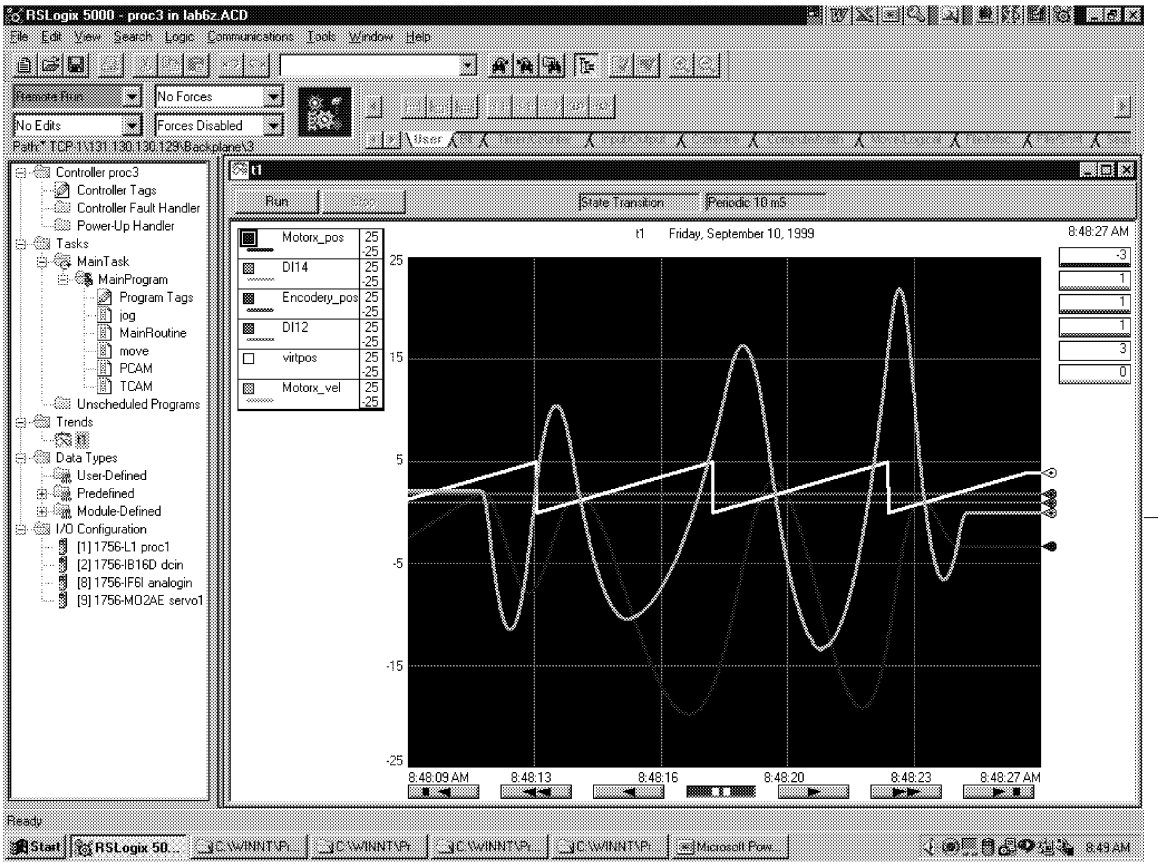
Consumed Axis

A Consumed Axis allows you to utilize produced/consumed tags to pass axis data between controllers. There also is capability to allow gearing and PCAM motion between axes owned by different Logix 5550 processors in the same chassis

A Motion Module is not required to create a Consumed Axis.

Graphical Trending Techniques

RSLogix 5000 offers the ability to monitor a view of any tag using RSTrendX . This includes the capability to monitor a CAM move, status of an input, etc.



Quiz Yourself

Take a few minutes to answer the following questions.

1. True or False: The 1756-M02AE servo module executes the ladder motion commands. It also creates position and velocity profile information.

2. RSLogix 5000 provides support for up to _____ types of motion commands.

3. _____ is used to complete axis configuration and motion programming support.

4. True or False: Each Logix5550 controller can support up to 32 servo axes.

5. The 1756-MO2AE servo card, for each axis, closes position/velocity loop updates in _____ µsec.

6. Name the instruction which allows you to slave to a master axis using a complex motion profile.

7. True or False: You need a motion axis card to create a virtual axis.

8. To graphically monitor a position axis tag for a CAM profile for instance, use the _____ feature.

Integrated Motion Answer Key

1. False. The Logix5550 does.

2. 30.

3. RSLogix 5000.

4. True.

5. 200

6. MAPC

7. False

8. RSTrendX

Check Your Configuration

Objective

- Identify the components of the ControlLogix demo unit.
- Prepare the system for hands-on exercises.

Demo Package Overview

The ControlLogix demo with motion is packaged into one unit. **Locate** the following components and check them off to indicate that you have identified them.

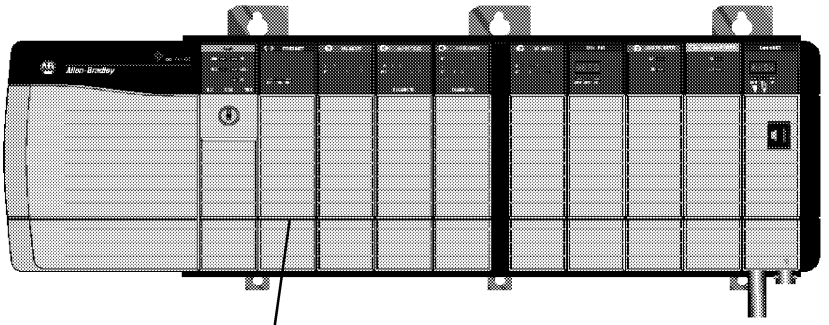
- ❑ Power on/off
- ❑ DC output card(s)
- ❑ DC input card
- ❑ Logix5550 controller
- ❑ ControlNet card
- ❑ ENET card
- ❑ Analog output card
- ❑ Analog input card
- ❑ 2 Axis Servo card
- ❑ Motor_x (associated Ultra 100) and Encoder_y wheels
- ❑ Digital inputs/outputs, push button/switches
- ❑ Analog potentiometers, AI0, AI1
- ❑ Analog output meters, AO0, AO1

Lab 1

Develop application to create Virtual Axis Routine

Objective

Review Appendix A, How to configure the ENET card
Review Virtual Axis Theory(See Appendix A to review how to build a virtual axis)
Develop an application program to JOG the Virtual Axis
Develop a GSV to monitor the Virtual Axis JOG



1756-L1 Processor Slot 3
Axes: VirtualL1

To give you maximum benefit of the workshop time, the following information has already been created in the program StartADVLab1.ACD

1756-L1 processor has been configured
1756-IB16D input card has been configured
The 1756-OB16D output card has been configured
1756-ENET Card has been configured

To start the lab, open StartADVLab1..ACD

Note: If you do not finish this lab in time, open file Lab1Final.ACD which includes all of the necessary information to complete the lab. Then go to the Lab Exercise to test the application.

Virtual Axis Overview

A virtual axis is an "internally" generated position reference. It has most of the attributes of a Servo axis except there is not a feedback or drive connection and it does not require a servo module.

* A virtual axis is capable of being used by all Motion Move type instructions including MAS, MAH, MAJ, MAM, MAG, MCD, MRP, MAPC, MATC.
* The number of virtual axes that can be supported is only limited by the Logix5550 maximum axis count of 32
* A virtual axis does not require a Servo Module
* A virtual axis does not require a MSO instruction to enable it

A virtual Axis can be used anytime internal position generation is required. The following might be examples.

As the Master Axis for a PCAM, TCAM, or Gear move
Use as a position reference to drive Position based outputs

An application example is a Horizontal Flow Wrapper:

Requirements

1. PCAM 3 axes to an infeed conveyor encoder.
2. Require the capability to run the 3 axes in the PCAM mode when the infeed conveyor is running and when the infeed conveyor is stopped

Solution

1. PCAM 3 axes to a virtual axis
2. Infeed running mode – gear the virtual axis to the infeed conveyor encoder
3. Infeed not running mode – jog the virtual axis

Step #1: If you have not opened the file StartADVLab1 do so now

Step #2: Virtual Axis, virtual_proc_1 Axis HAS BEEN CREATED. If you wish to create virtual_proc_1, open Appendix A and follow the directions.

Step #3: Create a New Routine for virtual_proc_1 Axis

Find Task in the Program Tree. Right click on Main Program, and select New Routine. Name it Virtual_proc_1. Click O.K. See Figure 1a.

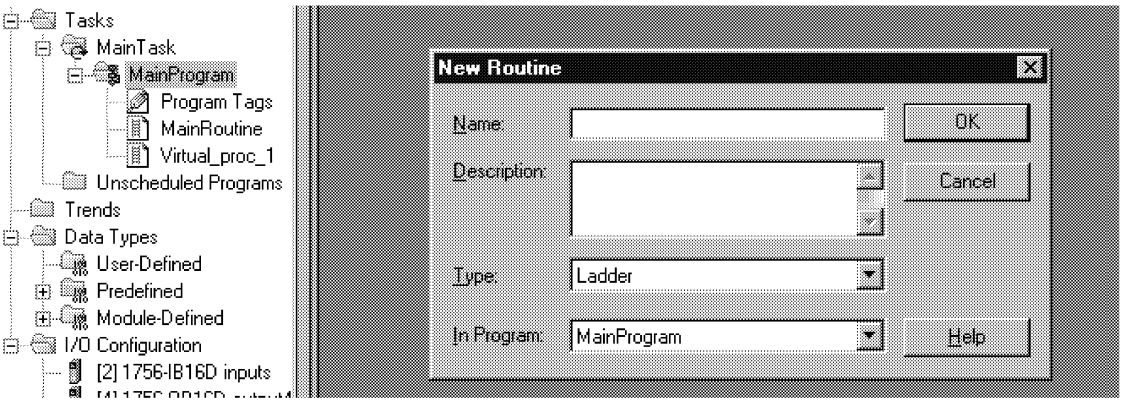


Figure 1a: Configure a New Routine

Step #4: Change the Main Routine

>>Open the Main Routine

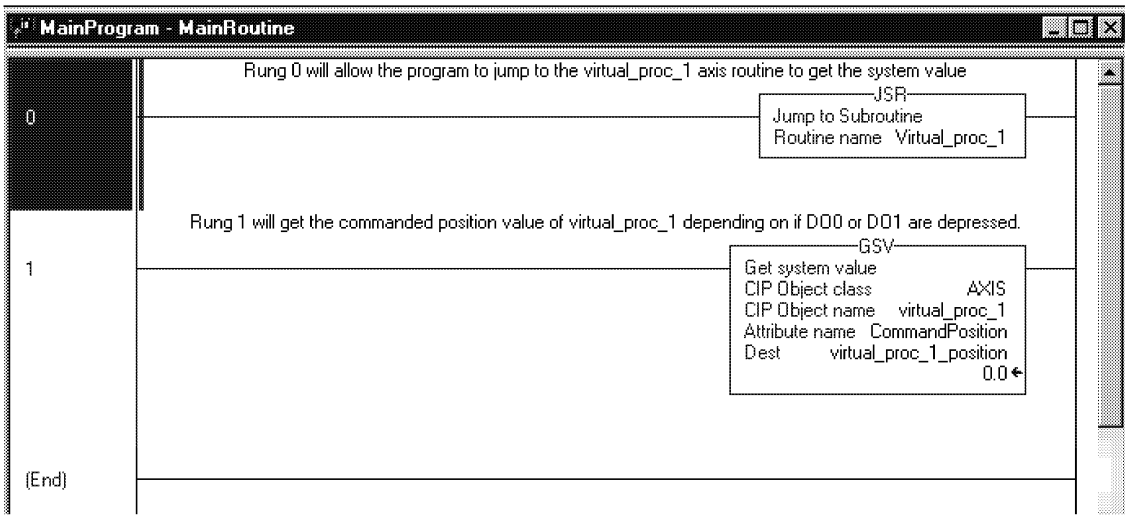

MainProgram - MainRoutine
Rung 0 will allow the program to jump to the virtual_proc_1 axis routine to get the system value
JSR
Jump to Subroutine
Routine name Virtual_proc_1
0
Rung 1 will get the commanded position value of virtual_proc_1 depending on if DO0 or DO1 are depressed.
GSV
Get system value
CIP Object class AXIS
CIP Object name virtual_proc_1
Attribute name CommandPosition
Dest virtual_proc_1_position
0.0
1
(End)

Figure 1b: Main Routine

>>Enter the two rungs in Figure 1b under Main Routine

>>JSR is located in the "Program Control" field in RSLogix 5000. See Figure 1c.

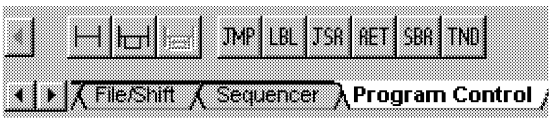


Figure 1c: Instruction Pallette

>>Note that Routine Name is empty. Double click on the ? in this field. Select
Virtual_proc_1 for this field.

>>You will notice two fields, Input and Return par. These fields are used to pass
parameters to and from this routine. We will not use these. Right mouse click in
each of these fields ?. Select “Remove Instruction Parameter” for each of these fields
such that you no longer have to enter any data.

>>To add the Rung 0 comment, select this rung by right mouse clicking on the Rung 0
and select “Edit Rung Comment”. Type in the comments and select the “check”
mark to accept this comment.

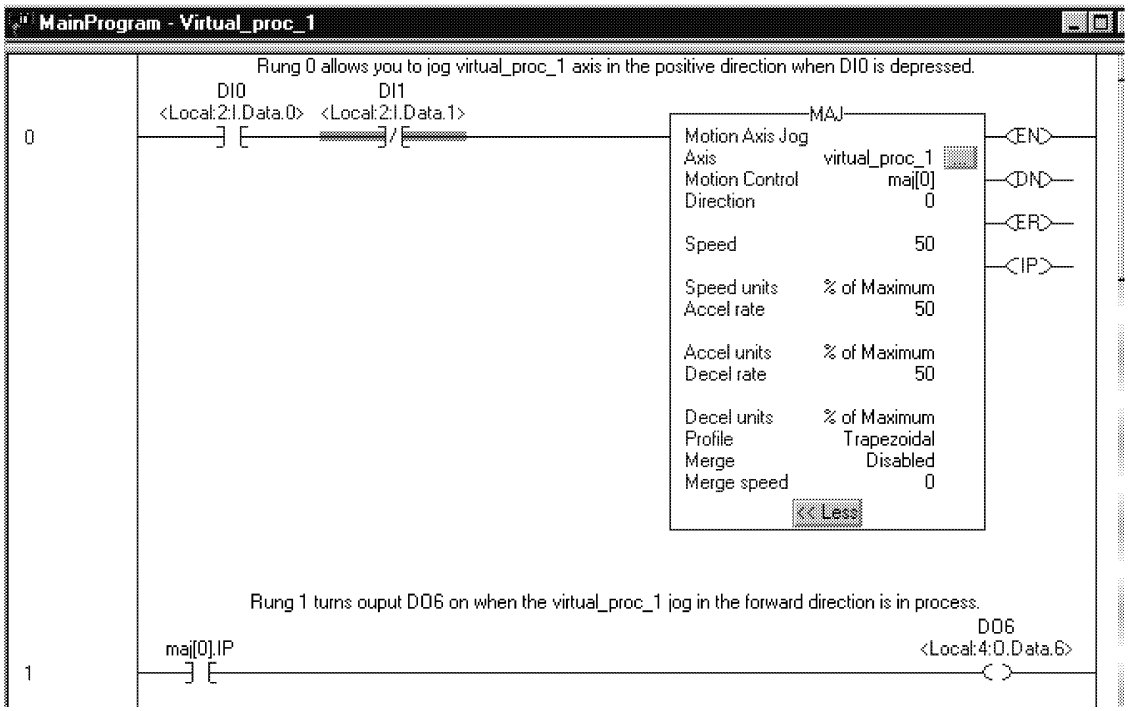
>>GSV is located in the “Input/Output” field in RSLogix 5000.

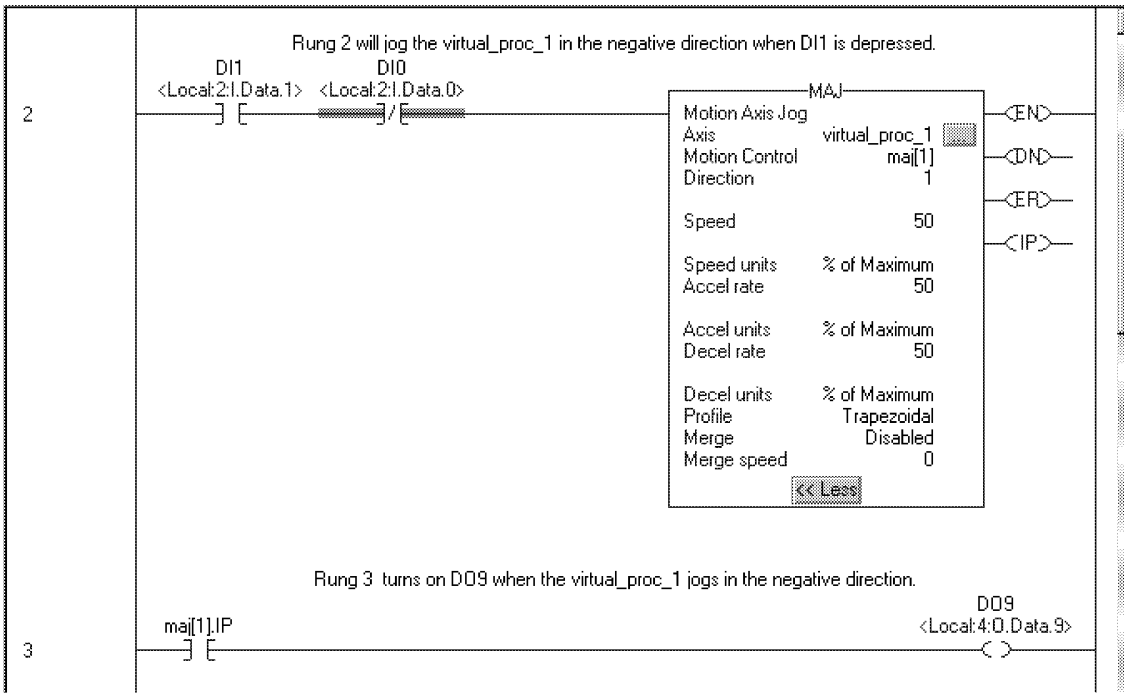
>>No tags need be created. They are created already for you. Simply place the
appropriate tag in the GSV field located in Figure 1b. To find the correct tag, double mouse click on the ? and search for the correct tag(Note that the tag names may be either in the “Program Scope” or the “Controller Scope” Tags folder)

Step #5:Enter the logic in the Virtual_proc_1 routine

>>Select the Virtual_proc_1 routine by double clicking on Virtual_proc_1.
>>Enter the Logic shown in Figure 1d.

HINT: All tags have already been created for you. To access simply place the cursor over the parameter field in the instruction, double click, select pull down, make sure to select controller scoped tags, select the tag. NOTE: SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE TAG NAME.


MainProgram - Virtual_proc_1
Rung 0 allows you to jog virtual_proc_1 axis in the positive direction when DI0 is depressed.
DI0
<Local:2:I.Data.0>
DI1
<Local:2:I.Data.1>
0
MAJ
Motion Axis Jog
Axis virtual_proc_1
Motion Control maj[0]
Direction 0
Speed 50
Speed units % of Maximum
Accel rate 50
Accel units % of Maximum
Decel rate 50
Decel units % of Maximum
Profile Trapezoidal
Merge Disabled
Merge speed 0
<< Less
EN
DN
ER
IP
Rung 1 turns ouput DO6 on when the virtual_proc_1 jog in the forward direction is in process.
maj[0].IP
DO6
<Local:4:O.Data.6>
1


Rung 2 will jog the virtual_proc_1 in the negative direction when DI1 is depressed.
DI1
<Local:2:I.Data.1>
DI0
<Local:2:I.Data.0>
2
MAJ
Motion Axis Jog
Axis virtual_proc_1
Motion Control maj[1]
Direction 1
Speed 50
Speed units % of Maximum
Accel rate 50
Accel units % of Maximum
Decel rate 50
Decel units % of Maximum
Profile Trapezoidal
Merge Disabled
Merge speed 0
<< Less
EN
DN
ER
IP
Rung 3 turns on DO9 when the virtual_proc_1 jogs in the negative direction.
maj[1].IP
DO9
<Local:4:O.Data.9>
3

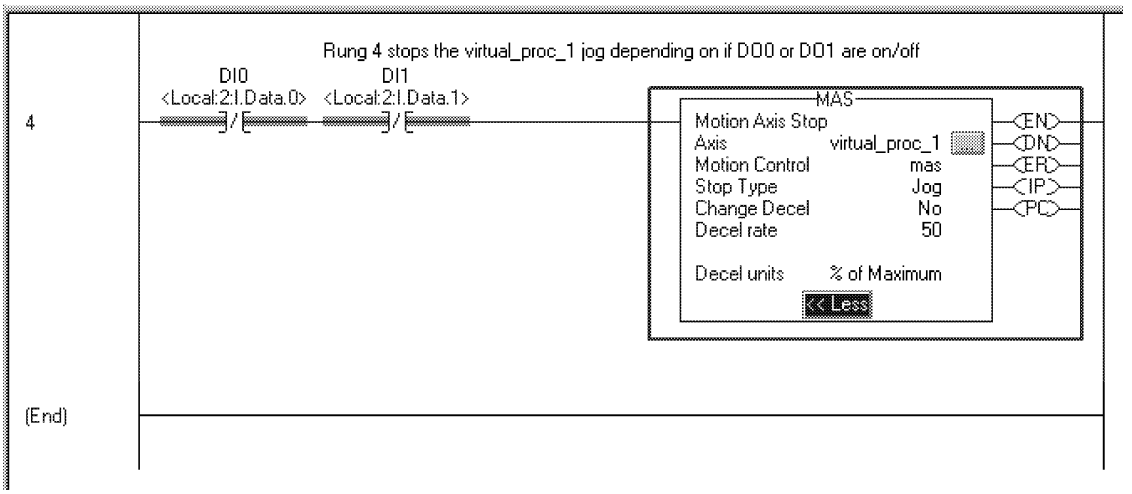


Figure 1d: Virtual_proc_1 Routine

>>The Examine On and Off as well as the Output Energize Instructions are located in the "User" Field.

>>The MAJ and MAS Instructions are located in the "Motion Move" Field.

Logic overview

>>Rung 0 - When DI0 is depressed the virtual axis Virtual_proc_1 Axis will jog in the forward direction

>>Rung 1 – When the MAJ (Motion Axis Jog) .IP bit is in process, Digital Output 6 will illuminate.

>>Rung 2 – When DI1 is depressed the Virtual_proc_1 Axis will jog in the reverse direction

>>Rung 3 – When the MAJ (Motion Axis Jog), .IP bit is in process, Digital Output 9 will illuminate.

>>Rung 4 – When DI0 or DI1 are not depressed, the Virtual_proc_1 Axis will stop.

Lab Exercise

1. Verify Program, Download (to Logix5550 processor in slot 1) and Run.). If you weren't able to download the file, select the Communications tab in RSLogix 5000. Go to RSWho. Expand the Ethernet, TCP-1 tree, via the backplane and choose the L1 processor in slot 1. Check "Apply Current Path to the Project". Select Apply. Select Download.
2. Toggle DI0 & DI1 inputs.
3. Monitor rung 1 of the Main Routine and note the position of axis, Virtual_proc_1, returned by the GSV instruction (in tag virtual_proc_1_position).
4. Select FILE, SAVE AS, and select a unique name and save this file.

Test your knowledge on the lab you have just completed!

1. What are the main differences between a servo and virtual axis?
2. What does the .IP bit from the MAJ(Motion Axis Jog) Instruction do?

Answers

1. The Servo is a physical axis (Drive, Motor, encoder). It requires assignment to a channel and a 1756-M02AE module. A virtual axis is an internally "generated" position Reference, there is not a physical AXIS associated with it and it does not require a servo module.

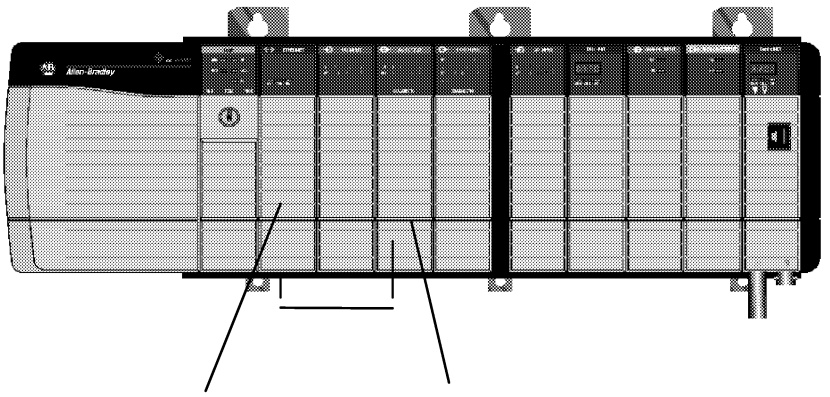
2. The .IP bit indicates the MAJ Instruction is “In Process”. You can sequence another instruction using this Motion Instruction bit as done here.

Lab 2

Create a Consumed Axis Associated with Processor L1 slot 1

Objective

Develop an application program to retrieve position information from Consumed1 axes



1756-L1 Slot 1
Axes: Virtual_proc_1

1756-L1 Slot 3
Axes: Consumed1

First minimize the Lab1Final.ACD that was created in Lab 1. Start another RSLogix 5000 session to use StartADVLab2.ACD. To give you maximum benefit of the workshop time, the following information has been already created in the program StartADVLab2.ACD. Look to Appendix B for the steps to create a consumed axis.

-1756-L1 processor has been configured

To start the lab, open StartADVLab2.ACD

Note: If you do not finish this lab in time, open Lab2Final which includes all of the necessary information to complete the lab. Then go to the Lab Exercise to test the application.

Consumed Axis Overview

The Logix 5550 controller supports the ability to produce (broadcast) and consume (receive) system shared TAGS. Produced data is accessible by multiple controllers over the control bus backplane or controlnet network.

The above is a description of the producer/consumer feature supported in ControlLogix. This feature has been expanded to allow TAGS of type "AXIS" to be used. This allows sharing of Axis information (position, status, registration, etc.) between multiple Logix 5550 controllers in the same chassis.

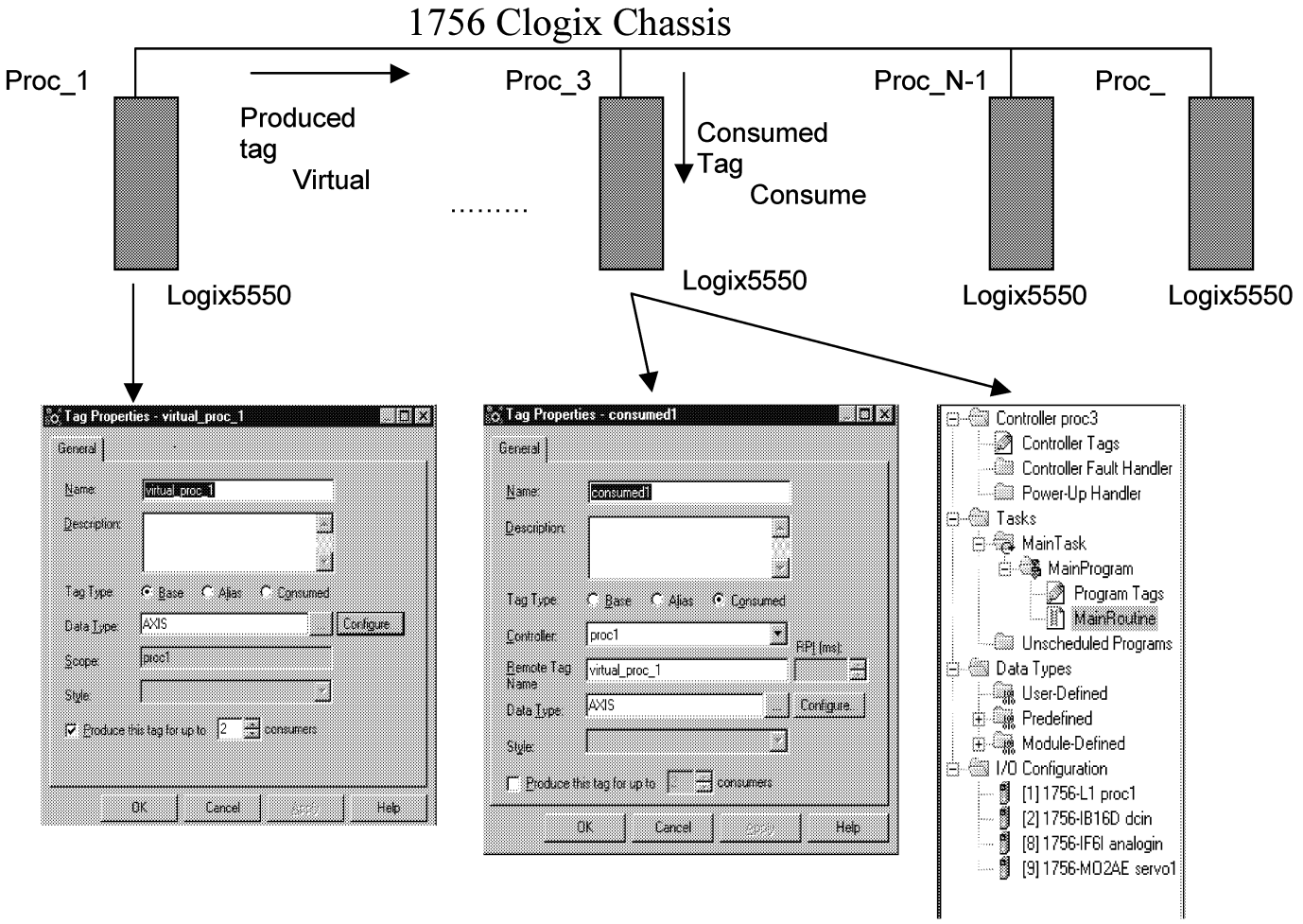
The Logix 5550 controller which "owns" the Axis can be setup to "Produce" the Axis information. One or more Logix 5550 controllers can then be configured to "consume" this information. Note the following:

>>Any Axis can be produced

>>A consumed axis is very similar to feedback only axis in terms of functionality

>>Built on the foundation of the Control Logix Producer-Consumer tag model

>>There is no limit to the number of consumed axes other than the 32 axes limit per Logix5550 controller



Step #1: If you have not done so already, Minimize the first Logix5000 program (currently running the Lab1Final.ACD program) and open a new Logix5000 session. Open the file StartADVLab2.

* Note: By opening a separate session for each Logix5550 processor in the chassis you can view and edit programs on-line and off-line for both processors at the same time.

Step #2: Enter the logic in the MainRoutine routine

>>Select the MainRoutine routine by double clicking on MainRoutine in the diagram Navigator

>>Enter the Logic shown in Figure 2 attached.

HINT: All tags have already been created for you. To access a tag, simply place the cursor over the parameter field in the instruction, double click, select pull down, make sure to select controller scoped tags, select the tag. NOTE: SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE TAG NAME

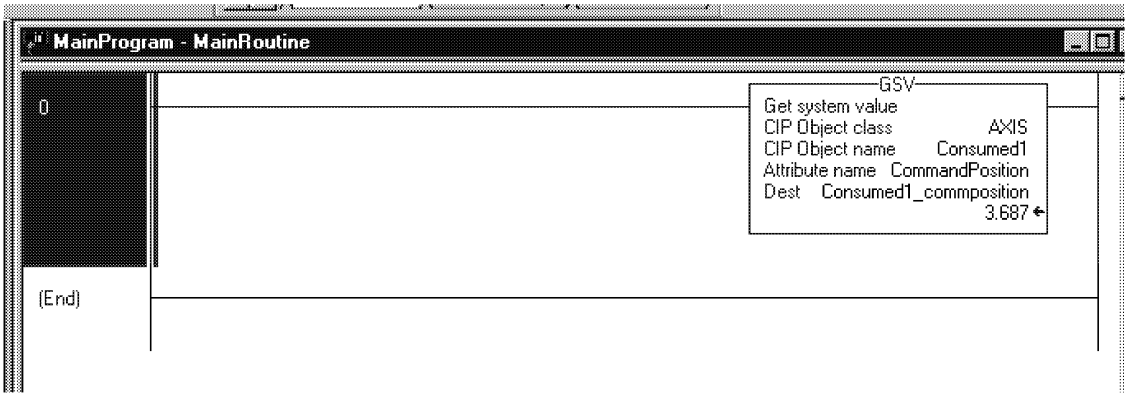


Figure 2: Main Routine For Consumed1 Axis GSV

The GSV instruction is in the input/output instruction tab.

Logic overview

>>*Rung 0 - The GSV blocks return position information for the specified axes. Commanded Position of Virtual_proc_1 (of Processor 1) will be produced such that it will be consumed in Processor 3. Notice you do not need a 1756-M02AE card for a consumed axis.*

Lab Exercise

- Verify Program, Download to processor in slot 3 and Run (Note program has already been setup for proper path using ENET connected to processor in slot 1). If you weren't able to download the file, select the Communications

tab in RSLogix 5000. Go to RSWho. Expand the Ethernet, TCP-1 tree, via the backplane and choose the L1 processor in slot 3. Check “Apply Current Path to the Project”. Select Apply. Select Download.

- Insure that the 1756-L1, slot1 processor is also in Run mode.
- Toggle DI0 & DI1 inputs. This will cause the virtual axis in proc_1 to jog. Note the Consumed1 axis position changes will reflect the position change (consumed1 is linked to Virtual_proc_1 axis in the slot1 processor).

Test your knowledge on the lab you have just completed!

1. How many different types of servo axes can be configured?
2. What must you do to an axis in processor x to make it consumed in processor y?
3. What does Consumed really mean?

Answers

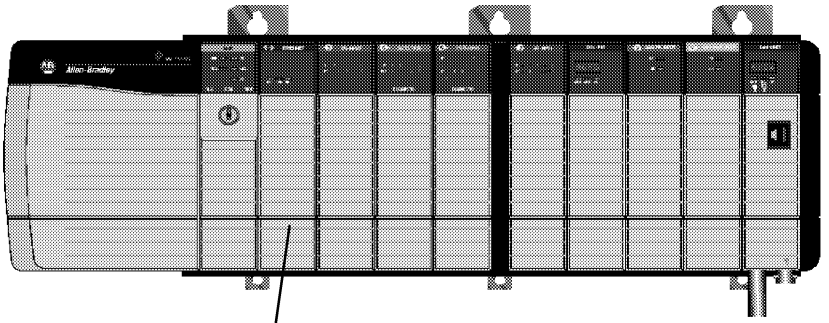
1. 4 – Servo, Position Only, Virtual and Consumed.
2. The axis in processor x must be produced on the Producer-Consumer Backplane.
3. Sharing axis information from processor to processor in a Control Logix Chassis.

Lab 3

Develop application to Gear Encoder_y to Motor_x, then Move while gearing is enabled

Objective

-- Develop an application program to Gear Encoder_y to Motor_x
-Develop a way to stop gearing
-Develop a way to move Motor_x while Encoder_y&Motor_x are still geared



1756-L1 Processor Slot 1
Axes: Encoder_y & Motor_x
Virtual_proc_1

To give you maximum benefit of the workshop time, the following information has already been created in the program StartADVLab3.ACD

1756-L1 processor has been configured
1756-IB16D input card has been configured
The 1756-OB16D output card has been configured
1756-M02AE Servo Card has been configured
1756-ENET Card has been configured

To start the lab, open StartADVLab3..ACD

Note: If you do not finish this lab in time, open file Lab3Final.ACD which includes all of the necessary information to complete the lab. Then go to the Lab Exercise to test the application.

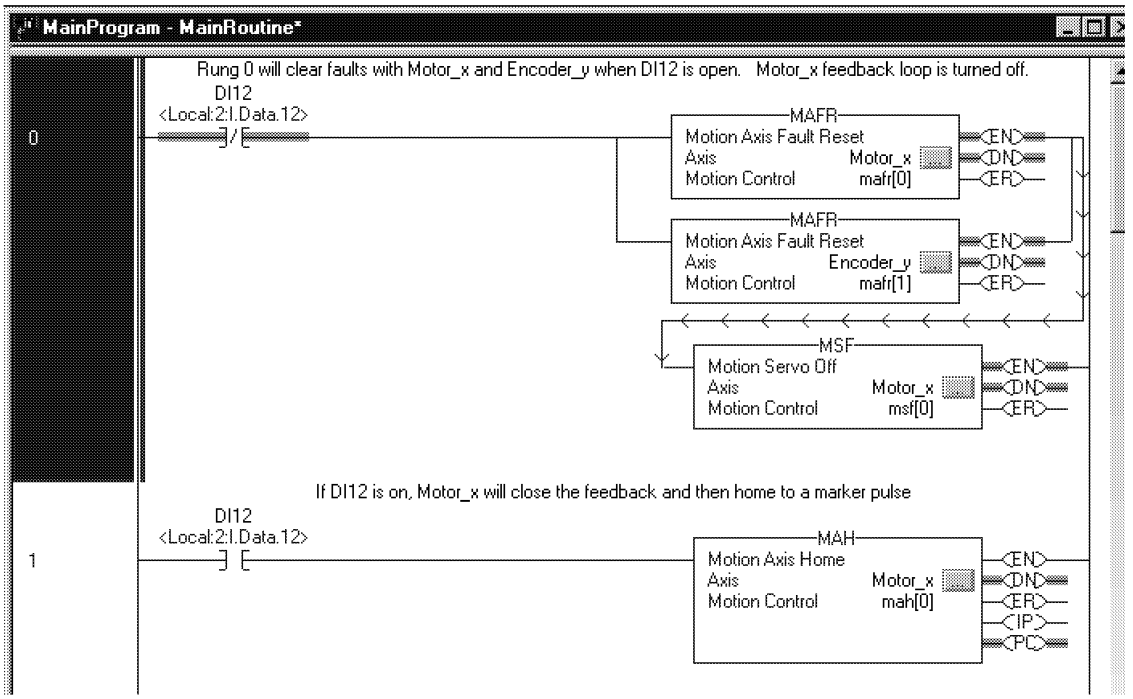
Step #1: If you have not opened the file StartADVLab3 do so now

Step #2: Create a New Routine for GearMoveAxis

Find Task in the Program Tree. Under Main Task, find Main Program. Right click to select a New Routine. Name it GearMove.

Step #3: Change the Main Routine

>>Open the Main Routine
>>Make the changes per Figure 3a attached. Do not add the rung comments.
>>MAFR and MSF are part of the "Motion State" Field in RSLogix 5000.
>>MAH is part of the "Motion Move" Field in RSLogix 5000.



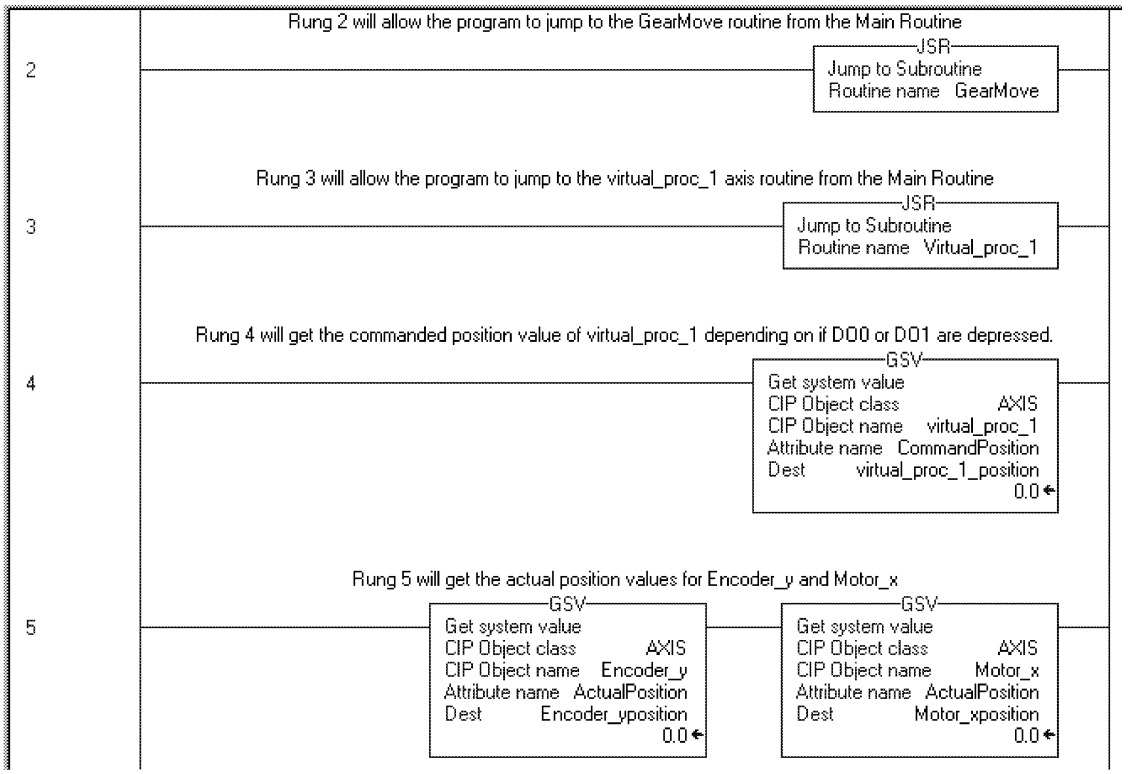
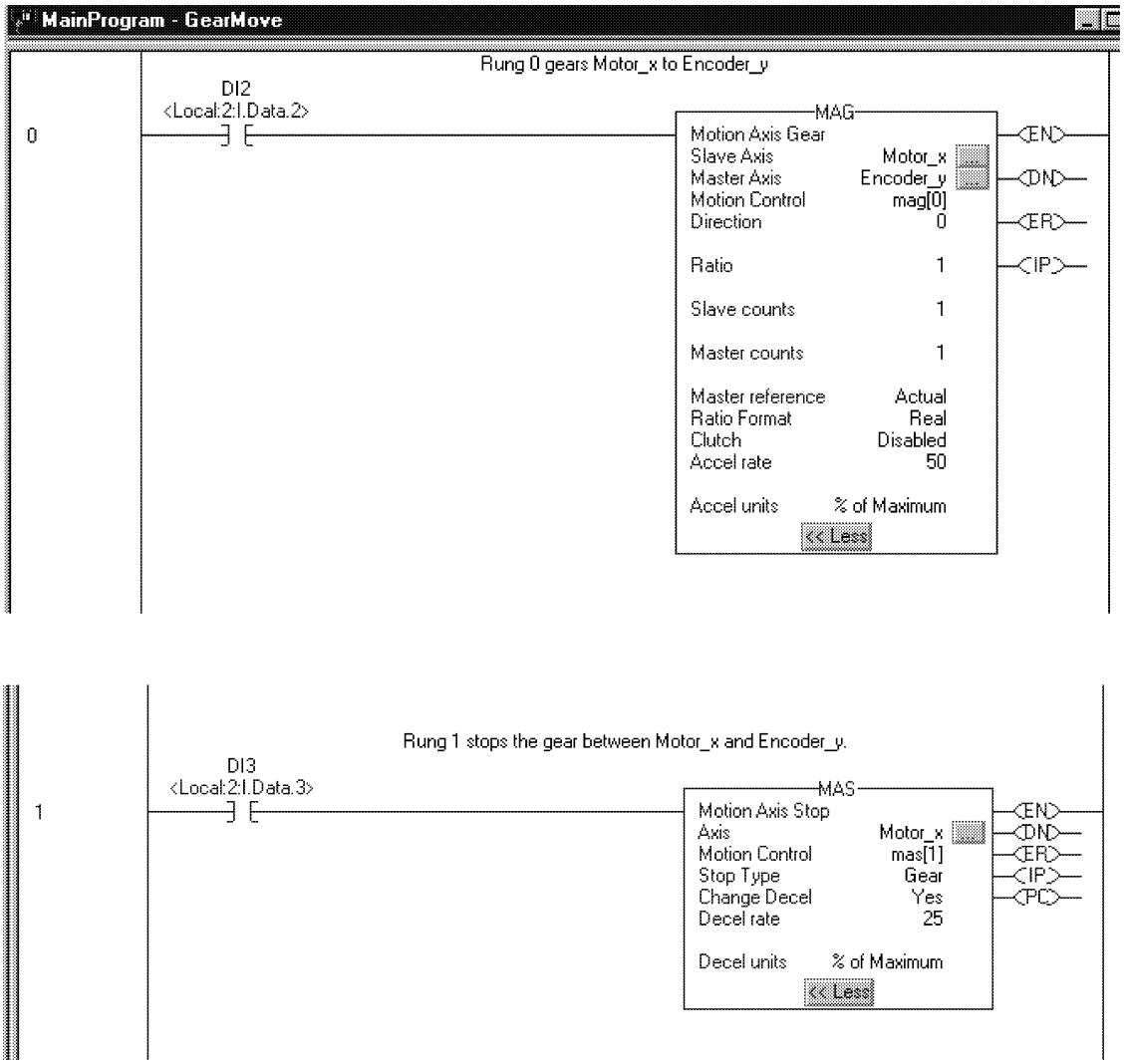


Figure 3a: Main Routine for Lab 3

Step #4:Enter the logic in the GearMove Routine

Select the GearMove routine by double clicking on GearMove
Enter the Logic shown in Figure 3b attached.

HINT: All tags have already been created for you. To access the tags, simply place the cursor over the parameter field in the instruction, double click, select pull down, make sure to select controller scoped tags, select the tag. NOTE: SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE TAG NAME. Check both the Controlled and Scoped Tags Folder.


MainProgram - GearMove
Rung 0 gears Motor_x to Encoder_y
DI2
<Local:2:I.Data.2>
0
MAG
Motion Axis Gear
Slave Axis Motor_x
Master Axis Encoder_y
Motion Control mag[0]
Direction 0
Ratio 1
Slave counts 1
Master counts 1
Master reference Actual
Ratio Format Real
Clutch Disabled
Accel rate 50
Accel units % of Maximum
<< Less
EN
DN
ER
IP
Rung 1 stops the gear between Motor_x and Encoder_y.
DI3
<Local:2:I.Data.3>
1
MAS
Motion Axis Stop
Axis Motor_x
Motion Control mas[1]
Stop Type Gear
Change Decel Yes
Decel rate 25
Decel units % of Maximum
<< Less
EN
DN
ER
IP
PC

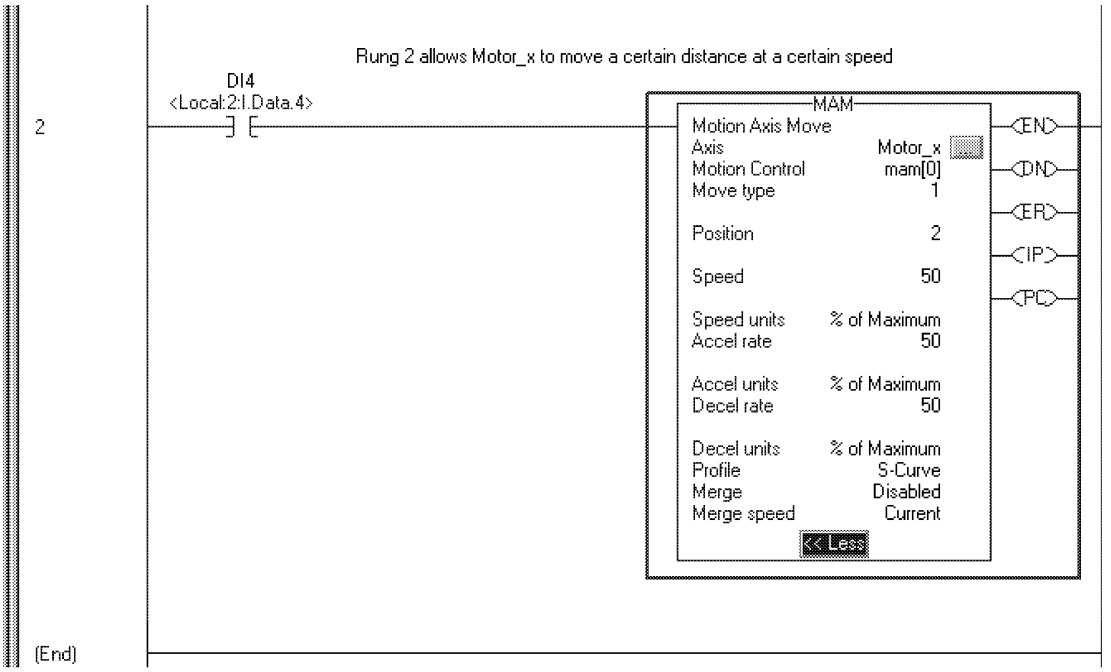


Figure 3b: Gear Move Routine for Lab 3

Logic overview

>>Rung 0 - When DI2 is depressed, Encoder_y will gear to Motor_x

>>Rung 1 – When DI3 is depressed, Encoder_y will no longer be geared to Motor_x

>>Rung 2- When DI4 depressed, Motor_x will move a certain distance at a certain speed. You can make an axis move while Encoder_y is geared to Motor_x.

Lab Exercise

1. Verify Program, Download (to Logix5550 processor in slot 1) and Run. Run (Note program has already been setup for proper path using ENET connected to processor in slot 1). If you weren't able to download the file, select the Communications tab in RSLogix 5000. Go to RSWho. Expand the Ethernet, TCP-1 tree, via the backplane and choose the L1 processor in slot 3. Check "Apply Current Path to the Project". Select Apply. Select Download.
2. Turn DI12 on to home the axis and turn feedback on.
3. Depress DI2 to gear Encoder_y to Motor_x. Spin Encoder_y to make sure Motor_x follows.
4. Depress DI3 to stop the gearing between Encoder_y to Motor_x.
5. Depress DI2 to once again gear Encoder_y to Motor_x. Open the Main Routine to review the GSV for Encoder_y and Motor_x. Move Encoder_y wheel and watch Motor_x follow. Simultaneously while moving Encoder_y, depress DI4 to watch Motor_x phase advance while Encoder_y is still geared.
6. Select FILE, SAVE AS, and select a unique name and save this file

Test your knowledge on the lab you have just completed!

1. MAG stands for what Motion Instruction?

2. If you want the Slave axis to move in the opposite direction of the Master Axis, what number should be entered in the Direction field of the MAG instruction?

Answers

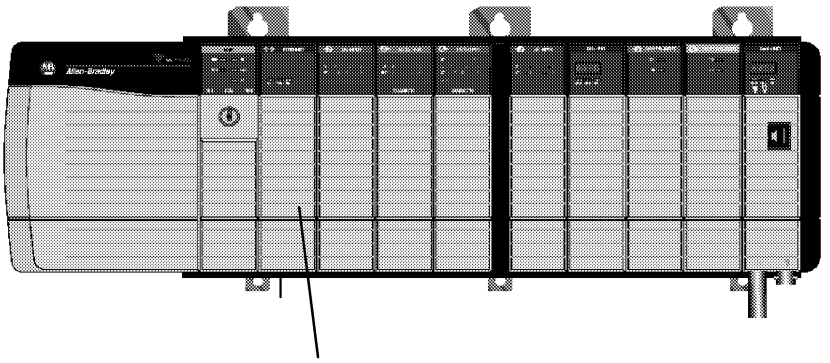
1. Motion Axis Gearing.
2. 1 = Slave Axis moves in the opposite direction of the Master Axis

Lab 4

Perform Position Cam Moves, Create CAM profiles using graphical CAM editor

Objective

- Build CAM Profiles using the RSLogix 5000 CAM Editor
- Use the MAPC Instruction to slave Motor_x to the following master inputs:
  1. Encoder_y
  2. Virtual1 Axis



1756-L1 Slot 3
Axes: Virtual_proc_1
Motor_x

Encoder_y

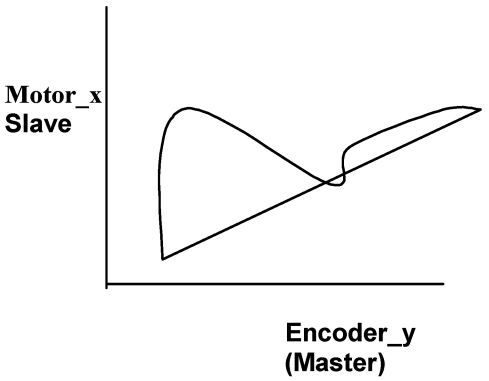
To give you maximum benefit of the workshop time a program shell has been provided for you to start with. The tags in this program shell have already been defined for you.

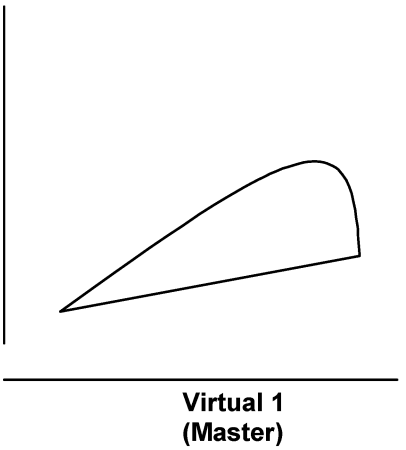
Note: If you do not finish this lab in time, open Lab4Final which includes all of the necessary information to complete the lab. Then go to the Lab Exercise to test the diagram.

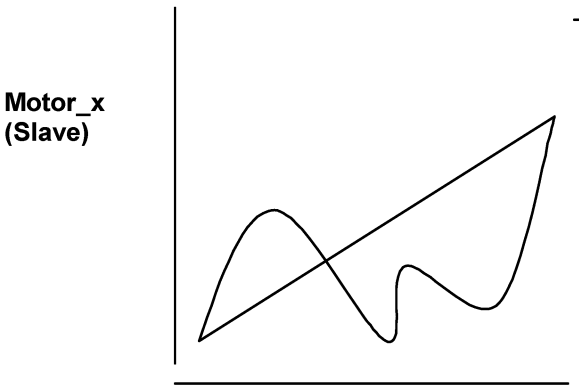
Motion Axis Position CAM – MAPC

The Motion Axis Position CAM function provides electronic camming between axes. Any axis can be used as the master including servo, feedback only, virtual, and consumed. The slave axis can be a virtual or servo axis.

The relationship between the master and slave axis is defined via a Tag of type CAM_PROFILE. This tag specifies a series of master positions, corresponding slave positions, and segment interpolation type (linear or spline).







Consumed_1(Master)

Creating the CAM profile

One of required fields in the MAPC instruction is “Cam Profile”. This field requires a TAG of type CAM_PROFILE. This TAG contains all the profile information which describes the relationship between the master and slave axes. The profile information contained in the “CAM_PROFILE” tag can be created in two ways:

1. Via Graphical CAM editor in RSLogix5000
2. Using the MCCP instruction in the application ladder

Method (1) is useful if the Profiles can be pre-defined and downloaded with the application program

Method (2) is useful if the Profile is created or modified during runtime via the application program

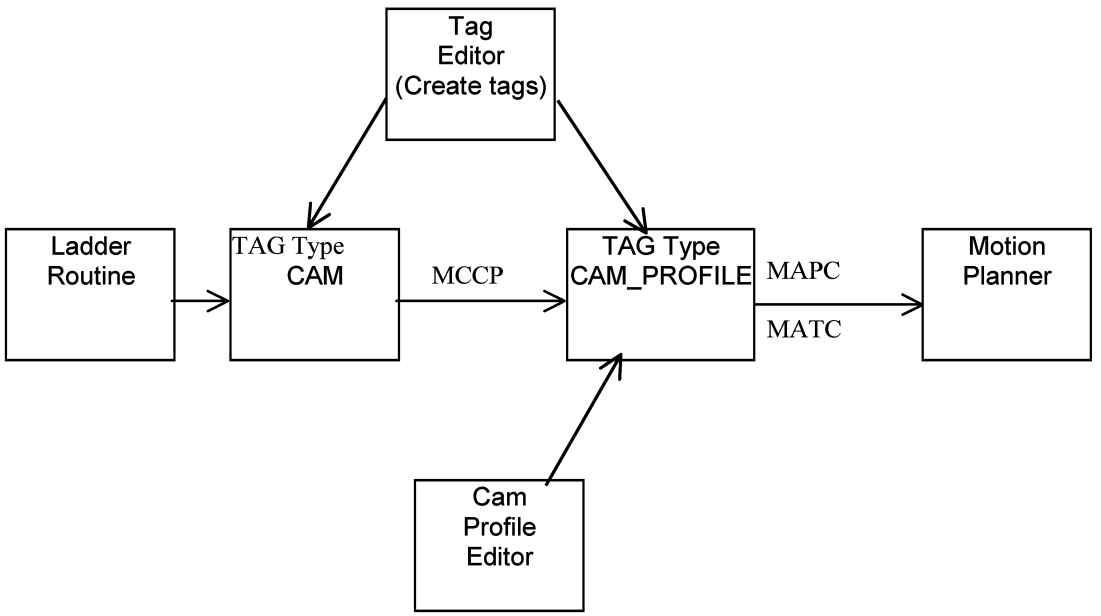
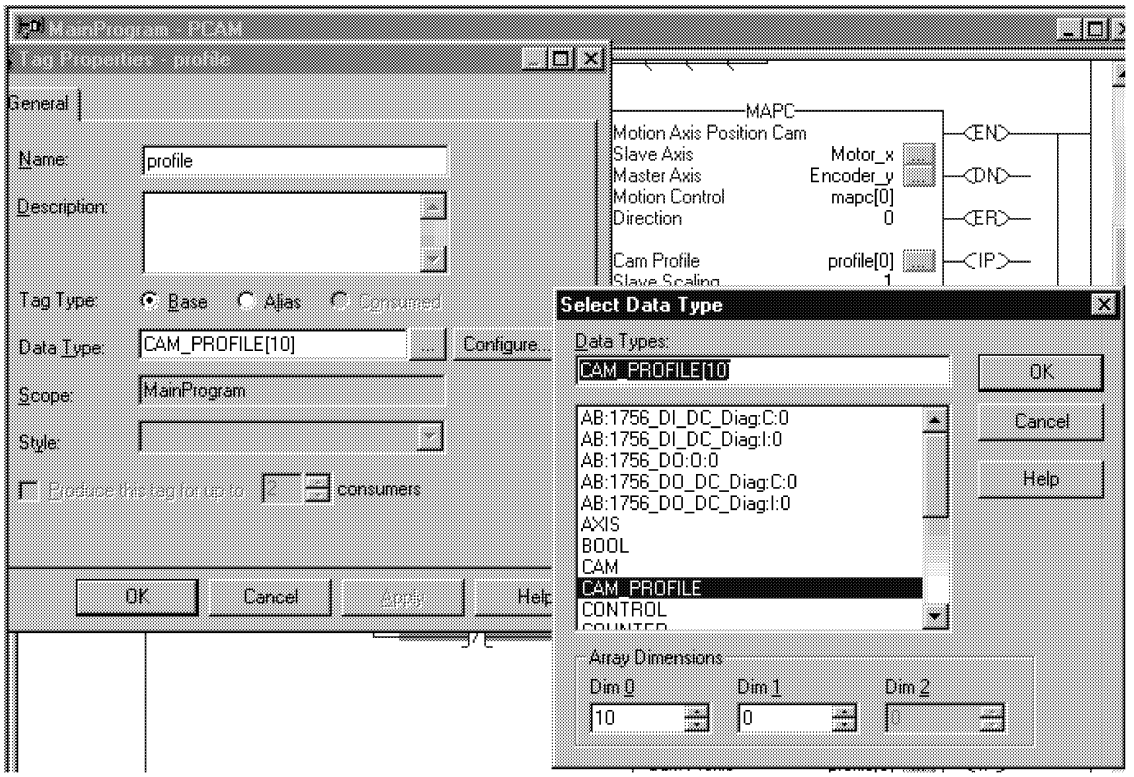


Figure 4 : ControlLogix CAM Operation

Create the Tag

The first step is to create the necessary tags. If the RSLogix5000 CAM editor will be used to create the CAM profile then a tag of type CAM_PROFILE is required for each profile to be created. If the MCCP instruction is used to create the CAM Profile then a Tag of type CAM is also needed. Both CAM_PROFILE and CAM tags should be specified as single dimension arrays with the array length set to match the number of CAM points required. The only limit to the number of profiles that can be created is available memory. Figure 4a below shows the dialog for creating a tag.

This tag has been named “profile”. It is a MAPC tag for the actual CAM Profile of 10 points . You will be responsible for entering the 10 points in the upcoming steps. This tag has been created. To create this tag simply right mouse click in the ? mark in the CamProfile field of the MAPC Instruction. Name the tag, “profile”. Under Data Type, select the box […] next to it. It is a



CAM_Profile type with an array Dim 0 of quantity 10.

Figure 4a – Creating CAM_PROFILE tag

Edit the CAM_PROFILE Tag

The profile information contained in the CAM_PROFILE type tag can be created/edited/monitored with a graphical CAM editor. This editor can be launched from the tag editor or from any instruction in which a tag of type CAM_PROFILE is used via a mouse click on the [...] box next to the tag. The CAM editor is shown in figure 4b. You will enter this data in the upcoming steps.

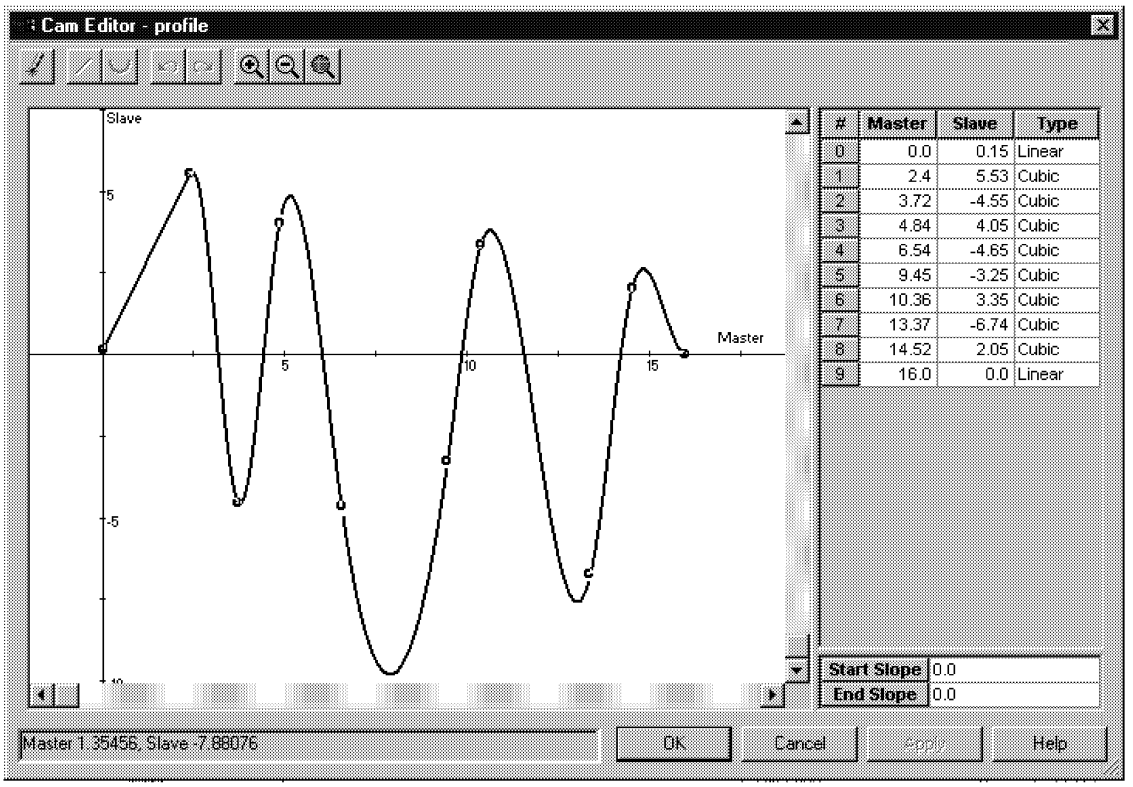


Figure 4b –CAM_PROFILE editor

MAPC instruction information

- Position CAMs are fully interpolated. Each point in the CAM array that was used to generate the CAM Profile can be configured for linear or cubic interpolation.
- Time Cam, Jog, incremental/absolute moves, and gearing moves can be superimposed on an active PCAM move.
- A position CAM Profile can be scaled in both master and slave dimensions.
- Blending of two position CAMs is possible without stopping motion by using the "Pending " feature
- Slaving to Commanded Position or actual position is supported.
- There is no limit to the number of axes that can be active in a CAM move
- Immediate, lock to master position (plus, minus, or any position transition) lockon options are supported
- It is possible to start at any point within the CAM _PROFILE. This helps facilitate fault recovery
- Motion Axis moves (such as incremental moves and phase shift moves) may be performed while CAMING to provide sophisticated phase and offset control while the slave axis is running.

Step #1: If you have not opened the file StartADVLab4 do so now

Step #2: Create the PCAM routine

>>Under Tasks, Main Program, right click and select new routine and name it PCAM.

Step #3: Change the Main Routine

>>Open the Main Routine

>>Insure to add the Logic shown in Figure 4c attached. You need only add the JSR to the PCAM routine rung. Add the rung after the JSR to the Virtual_proc_1 rung (rung 3). Note, again, do not worry about the rung comments.

HINT: All tags have already been created for you. To access simply place the cursor over the field in the instruction field, double click, select pull down, select controller scoped tags, (if not already selected) select the proper tag. SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE NAME OF THE TAG.

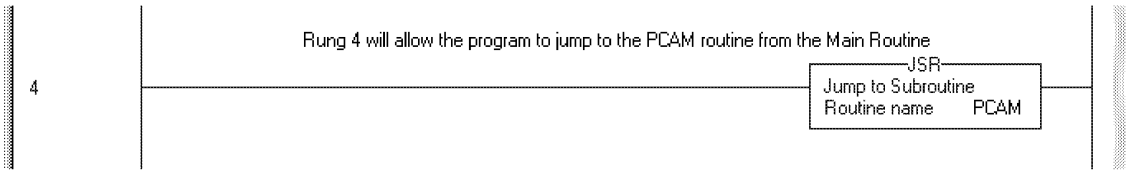


Figure 4c: Main Routine addition of the JSR to PCAM Routine

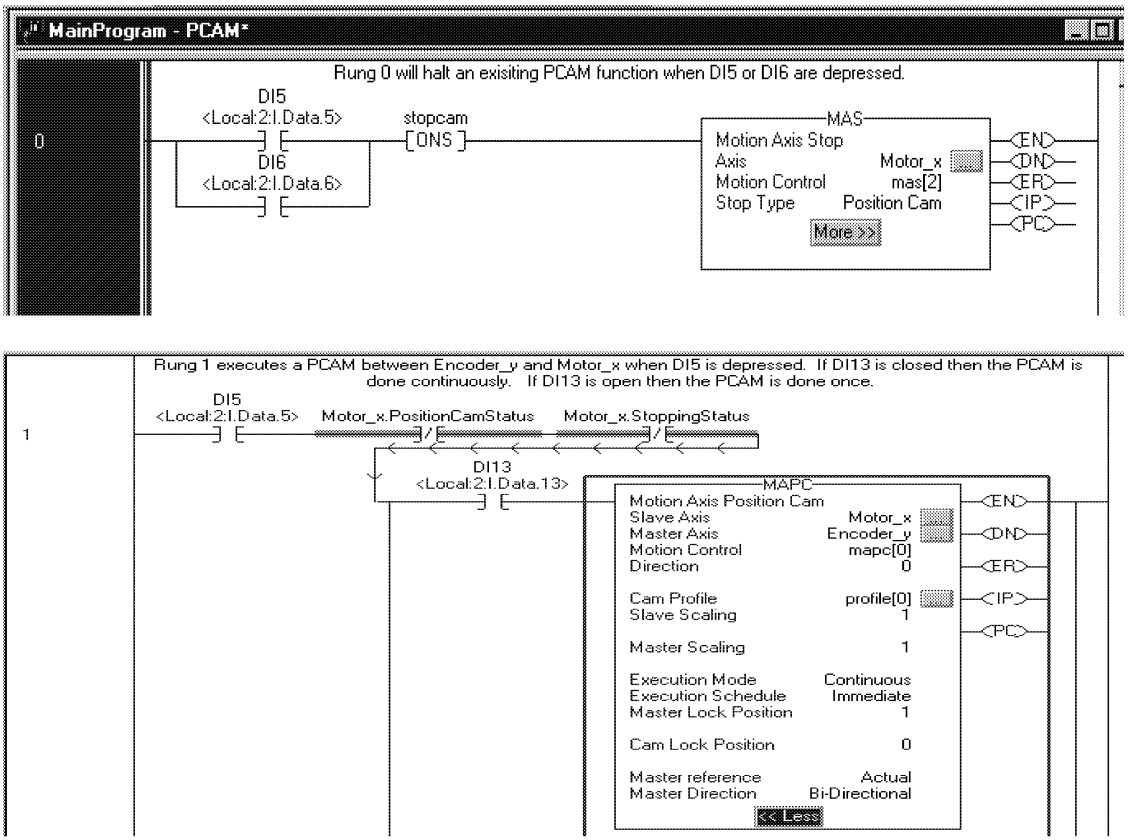
MainRoutine Logic overview

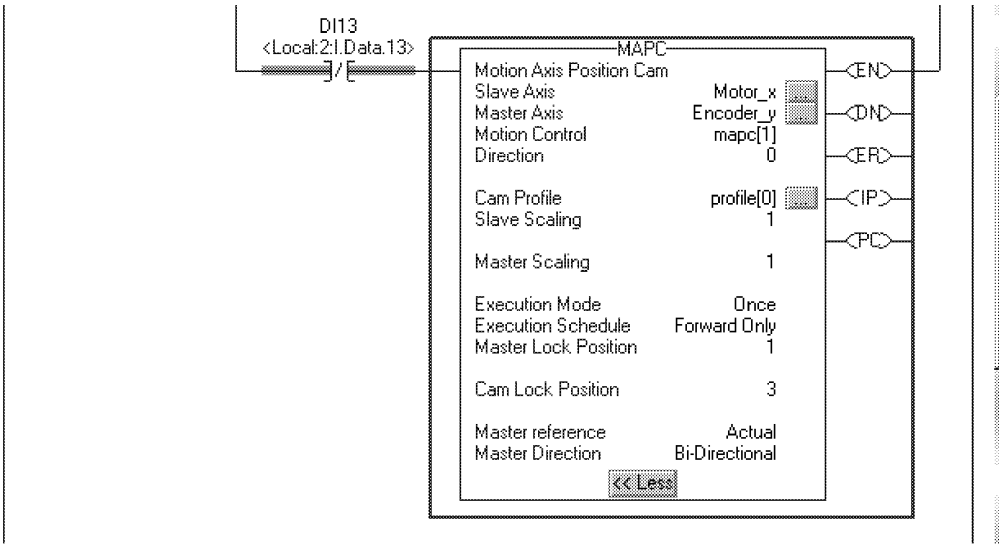
Rung 0 – Turn DI12 off to reset faults and disable Motor_x
Rung 1 – Enable and home Motor_x by turning DI12 on.
Rung 2 – Will Jump to GearMove Routine
Rung 3- Jump to Virtual_proc_1 Routine
Rung 4 – Jump to PCAM Routine
Rung 5 – GSV block to monitor Virtual_proc_1 axis.
Rung 6 – GSV block to monitor Encoder_y and Motor_x actual position

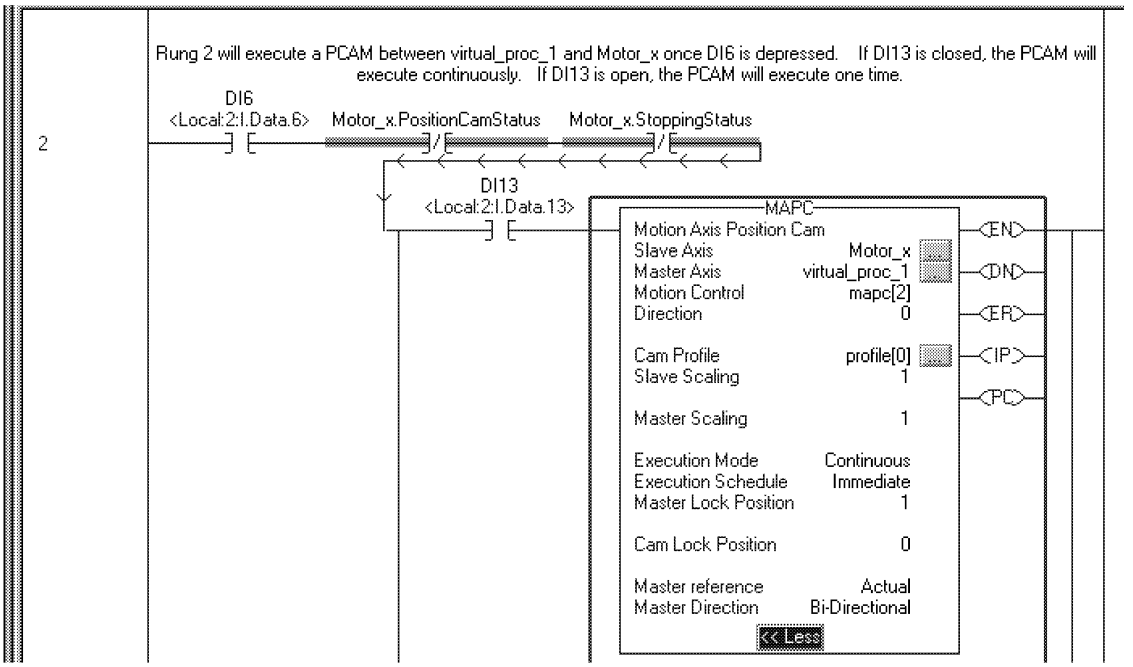
Step #4: Add Logic to the PCAM routine

>>Select the PCAM routine by double clicking on PCAM in the diagram Navigator
>>Add the logic shown in figure 4d.

HINT: All tags but the CAM profile have already been created for you. To access simply place the cursor over the field in the instruction field, double click, select pull down, select controller scoped tags, (if not already selected) select the proper tag. SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE NAME OF THE TAG.




DI13
<Local:2:I.Data.13>
MAPC
Motion Axis Position Cam
Slave Axis Motor_x
Master Axis Encoder_y
Motion Control mapc[1]
Direction 0
Cam Profile profile[0]
Slave Scaling 1
Master Scaling 1
Execution Mode Once
Execution Schedule Forward Only
Master Lock Position 1
Cam Lock Position 3
Master reference Actual
Master Direction Bi-Directional
<< Less
EN
DN
ER
IP
PC


Rung 2 will execute a PCAM between virtual_proc_1 and Motor_x once DI6 is depressed. If DI13 is closed, the PCAM will execute continuously. If DI13 is open, the PCAM will execute one time.
2
DI6
<Local:2:I.Data.6>
Motor_x.PositionCamStatus
Motor_x.StoppingStatus
DI13
<Local:2:I.Data.13>
MAPC
Motion Axis Position Cam
Slave Axis Motor_x
Master Axis virtual_proc_1
Motion Control mapc[2]
Direction 0
Cam Profile profile[0]
Slave Scaling 1
Master Scaling 1
Execution Mode Continuous
Execution Schedule Immediate
Master Lock Position 1
Cam Lock Position 0
Master reference Actual
Master Direction Bi-Directional
<< Less
EN
DN
ER
IP
PC

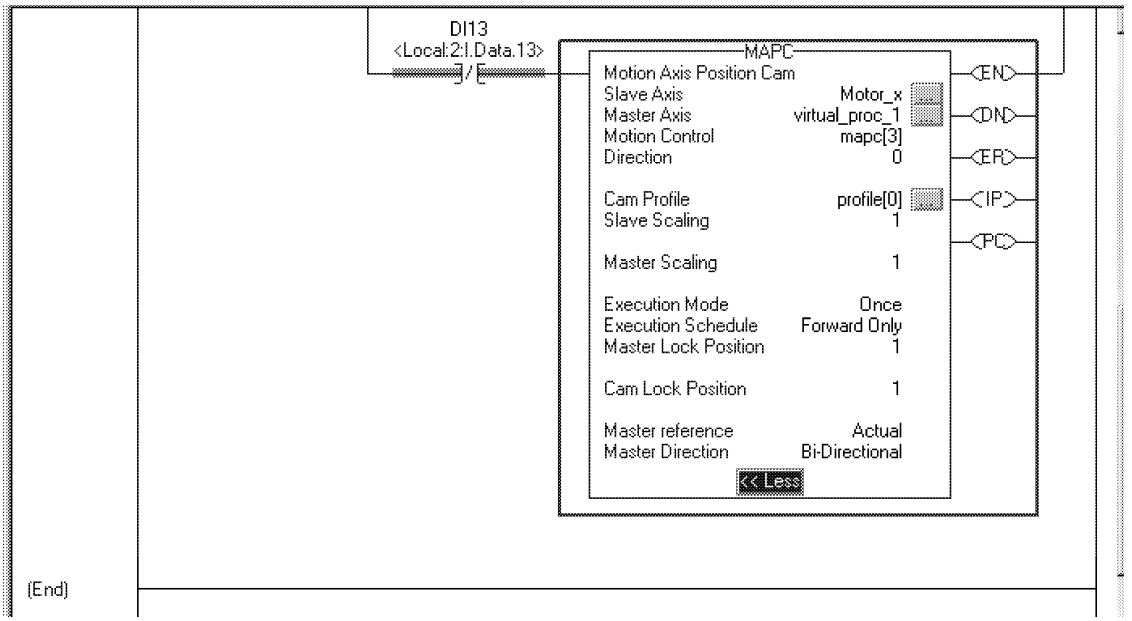


Figure 4d: PCAM Routine

Step #5: Edit the CAM_Profile Tag – The profile information contained in the CAM_Profile type tag can be created/edited/monitored with a graphical CAM editor. This editor can be launched from the tag editor or from any instruction in which a tag of type CAM_Profile is used via a mouse click on the [...] box next to the tag "profile" in our case. See Figure 4b as a reminder.

* Select the […] button next to the profile tag
* You should have an empty CAM Editor for profile
*Select the upper left most button to enter a data point (remember you have scaled your CAM points to 10 max in the step 4 above).
* Enter the following 10 points in the CAM Table. Use the tab or arrow key to get to the proper table location.

Point	Master	Slave	Type
0	0	0.15	Linear
1	2.4	5.53	Cubic
2	3.72	-4.55	Cubic
3	4.84	4.05	Cubic
4	6.54	-4.65	Cubic
5	9.45	-3.25	Cubic
6	10.36	3.35	Cubic
7	13.37	-6.74	Cubic

8	14.52	2.05	Cubic
9	16.00	0	Linear

- Start Slope = 0.00
- End Slope = 0.0
- Select Apply
- Hit OK

Another way to enter data is to build a table such as an Excel File and import it. We will do this in Lab 5. Finally, the CAM graphical editor can be used to change any of the points. Select the [...] button next to the profile tag. Left mouse click on and hold point, (16,0) and proceed to alter the final end point of the curve. Notice the table changes with the graphical editor. If you do alter this table, move the points back to the above table or just re-enter the values before proceeding.

Create Trends

Double mouse click on the Trend folder in RSLogix 5000. Create a new trend chart called PCAM and hit OK.

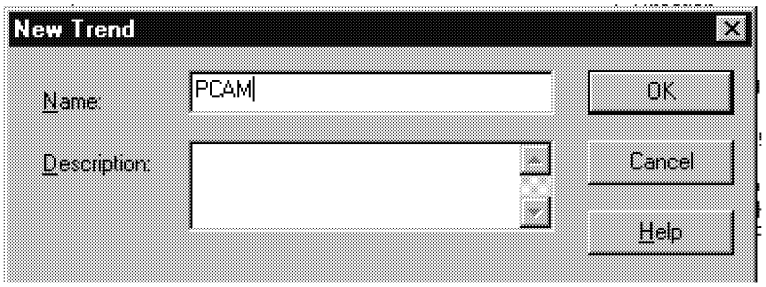


Figure 4E: Create a Trend

Double click on the PCAM folder under Trends. Notice there are no tags in the Trends Section. Put the mouse pointer on the chart and then right click. Select Chart Properties. Select Folder Pens. Notice no tags are configured here. These tags can be configured to the PCAM chart to continuously monitor the history of the tags. In other words, we can monitor axis position or the setting of a switch, etc.

Select Add/Configure Tags. Select the Scope for the Main Program. Select Encoder_yposition tag and hit Add. Do the same for Motor_xposition and virtual_proc_1_position. Hit Apply and then OK.

We will monitor these tags once we are online by hitting Run. We will do this in the Exercise section.

PCAM routine logic overview

>>Rung 0 - stops active cam move.

>>Rung 1 - When DI5 is pressed a PCAM move with master axis Encoder_y and slave axis MOTOR_X will start. If DI13 is on the mode will be continuous and lock will be immediate. If DI13 is off the mode will be once only and the lock will be on forward transition through Encoder_y position of 1.0. The tag Profile contains the profile information and determines the relationship between the master and slave axis.

Spin Encoder_y by hand to view the PCAM action.

>>Rung 2 – When DI6 is pressed a PCAM move with master axis virtual_proc_1 and slave axis MOTOR_X will start. If DI13 is on the mode will be continuous and lock will be immediate. If DI13 is off the mode will be once only and the lock will be on forward transition through virtual_proc_1 position of 1.0. The tag Profile contains the profile information and determines the relationship between the master and slave axis. Depress DI0 or DI1 to move the virtual_proc_1 axis such that the slave follows based on the CAM Profile.

Lab Exercise

1. Verify, Download and Run the PCAM Program. Run (Note program has already been setup for proper path using ENET connected to processor in slot 1). If you weren't able to download the file, select the Communications tab in RSLogix 5000. Go to RSWho. Expand the Ethernet, TCP-1 tree, via the backplane and choose the L1 processor in slot 3. Check "Apply Current Path to the Project". Select Apply. Select Download.
2. Turn Switch DI12 to home the Motor_x Axis
3. Turn Switch DI13 on. Depress DI5. Spin Encoder_y CW/CCW a few times and watch Motor_x reaction. Repeat step 3 with DI13 off and note the difference.
4. Turn Switch DI13 on. Depress DI6. Toggle DI0 & DI1 to jog the virtual_proc_1 Axis CW/CCW and watch Motor_x reaction. Repeat step 4 with DI13 off and note the difference.
5. Double left mouse click on Trends in the Program Tree. You will see a PCAM folder already created. Double click on PCAM. You will see 3 tags already configured (virtual_proc_1_position, Motorx_position and Encoder_yposition).
6. Hit the Run button in the Trends Field.
7. Toggle Switch DI12 to re-home the axis.
8. Go back and run through step 3. Notice the changing in virtual_proc_1_position and Motorx_positions on the Trend Screen. Spin the Encoder_Yposition and watch the position changes.
9. If any Logix 5550 Motion Card Fault should occur, simply toggle the DI12 Switch and restart the program (make sure Motor_x Feedback is on).

Test your knowledge on the lab you have just completed

1. Is there a limit to the number of axes that can be active in a CAM move?

2. What types of CAM profiles are available?

3. What is the difference between using CAM editor and the MCCP instruction?

4. Is there a limit of CAM points available?

Answers

1. No, there is no limit

2. Position CAM and Time CAM

3. CAM editor is used when profiles are pre-defined and downloaded with the program. MCCP is used if the profile is being created or modified during runtime of the application program.

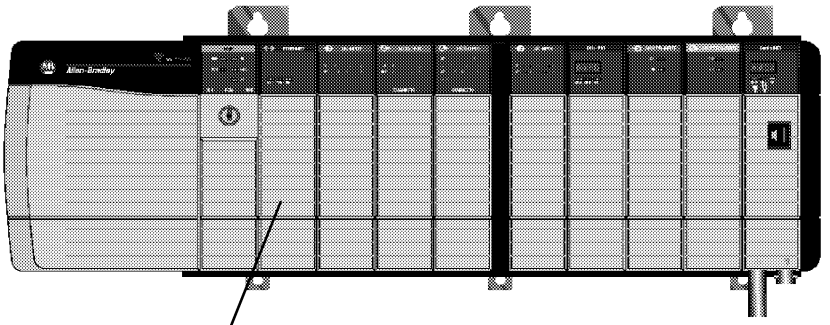
4. CAM point limitation is based on available processor memory.

LAB 5

Perform Time Cam Moves, Create CAM profiles using the MCCP instruction

Objective

- Build CAM Profiles using the MCCP instruction
- Build an Excel File and import it to the MCCP instruction
-Use the MATC Instruction to perform a time based CAM move



1756-L1 Slot 1
Axes: virtual_proc_1
Motor_x
Encoder_y

To give you maximum benefit of the workshop time a program shell has been provided for you to start with. The tags in this program shell have already been defined for you.

Note: If you do not finish this lab in time, open Lab5Final which includes all of the necessary information to complete the lab. Then go to the Lab Exercise to test the diagram.

Motion Axis Time CAM – MATC

The Motion Axis Time CAM function is very similar to the Motion Axis Position CAM – MAPC command. The main difference is that TIME is used as the master axis instead of a position reference.

MATC instruction information

Time CAMs are fully interpolated. Each point in the CAM array that was used to generate the CAM Profile can be configured for linear or cubic interpolation.

Position Cam, Jog, incremental/absolute moves, and gearing moves can be superimposed on an active TCAM move.
A time CAM Profile can be scaled in both master and slave dimensions.
Blending of two time CAMs are possible without stopping motion by using the "Pending feature
There is not limit to the number of axes that can be active in a CAM move

Step #1: If you have not opened the file StartADVLab5 do so now

Step #2: Create the new routine as TCAMwith MCCP.

>>Under Tasks, Main Program, right click and select new routine and name it TCAMwithMCCP. Select OK.

Step #3: Add additional logic in the MainRoutine routine

Select the MainRoutine routine by double clicking on MainRoutine in the diagram Navigator
Insure the Logic matches Figure5a. Put this JSR rung after the PCAM rung (rung 4).

HINT: All tags have already been created for you. To access simply place the cursor over the field in the instruction field, double click, select pull down, select controller scoped tags, (if not already selected), select the proper tag. SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE NAME OF THE TAG.

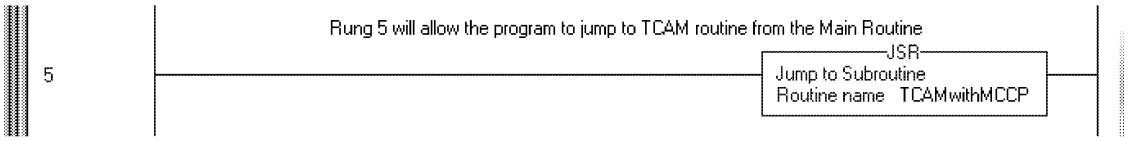

Rung 5 will allow the program to jump to TCAM routine from the Main Routine
5
JSR
Jump to Subroutine
Routine name TCAMwithMCCP

Figure 5a: Main Routine of adding a JSR for the TCAM routine

MainRoutine Logic overview

Rung 0 – Turn DI12 off to reset faults and disable Motor_x
Rung 1 – Enable and home Motor_x by turning DI12 on.
Rung 2 – Will Jump to GearMove Routine
Rung 3- Jump to Virtual_proc_1 Routine
Rung 4 – Jump to PCAM Routine
Rung 5- Jump to TCAM Routine
Rung 6 – GSV block to monitor Virtual_proc_1 axis.
Rung 7 – GSV block to monitor Encoder_y and Motor_x actual position

Step #4: Add Logic to the TCAM routine

Select the TCAM routine by double clicking on TCAMwithMCCP folder in the diagram Navigator
Add the logic shown in figure 5b below

HINT: All tags have already been created for you. To access simply place the cursor over the field in the instruction field, double click, select pull down, select controller scoped tags, (if not already selected), select the proper tag. SELECT THE TAG FROM THE SCROLL LIST, DO NOT TYPE IN THE NAME OF THE TAG.

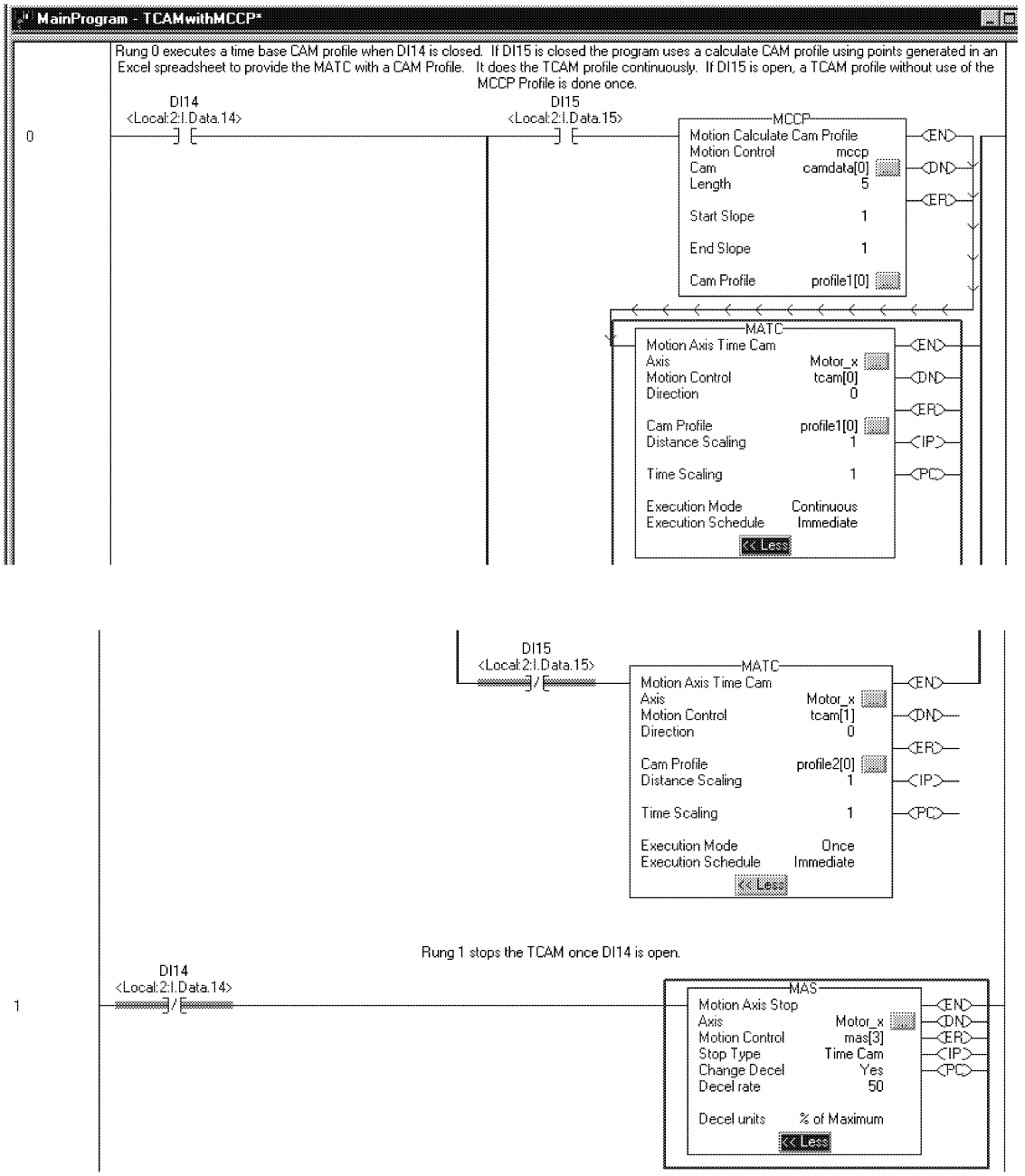


Figure 5b: TCAM with MCCP instruction Routine

TCAM routine logic overview

Rung 0 – When DI14 is enabled a TCAM move on Motor_x will start. If DI15 is set the TCAM will execute in the continuous mode and profile information contained in the tag CAMDATA (X pos, Y pos, segment type) will be used in a MCCP call to generate the profile information in tag PROFILEs. The data will be entered with an Excel table of 5 points. The tag Profile1 contains the profile information and determines the relationship between time (master axis)

and Motor_x. Note that even though the MCCP call is used to create the data in the Tag Profile1, the profile can still be viewed with the CAM editor.

If DI15 is off when DI14 is enabled then the TCAM will execute in the once mode and the tag PROFILE2 will be used to defined the relationship between time (master axis) and Motor_x. Profile2 has been created for you

Rung 1 – stops active TCAM move

Editing data in CAMDATA

When switch DI15 is set and DI14 is switched on the profile information contained in the tag CAMDATA (Xpos Y pos, segment type) will be used in a MCCP call to generate the profile information in tag PROFILE1. By modifying the data in the CAMDATA the information in PROFILE1 will be changed the next time DI14 is switched on. The data in CAMDATA can be modified using the tag monitor.

Let us use an Excel Spreadsheet to create a CAMDATA profile of 5 points. See Figure 5c.

0	0.15	0
2	-2	0
4	2	0
6	6	0
10	0	0

Figure 5c : Excel Spreadsheet of 5 points for MCCP Instruction

>>Hi-lite the 5 data points, not including the titles of the Excel Spreadsheet. Copy this array of items. Note, for type, 0 = Cubic and 1 = Linear. These values are interpreted by RSLogix 5000 as the appropriate Cubic or Linear Curve type.

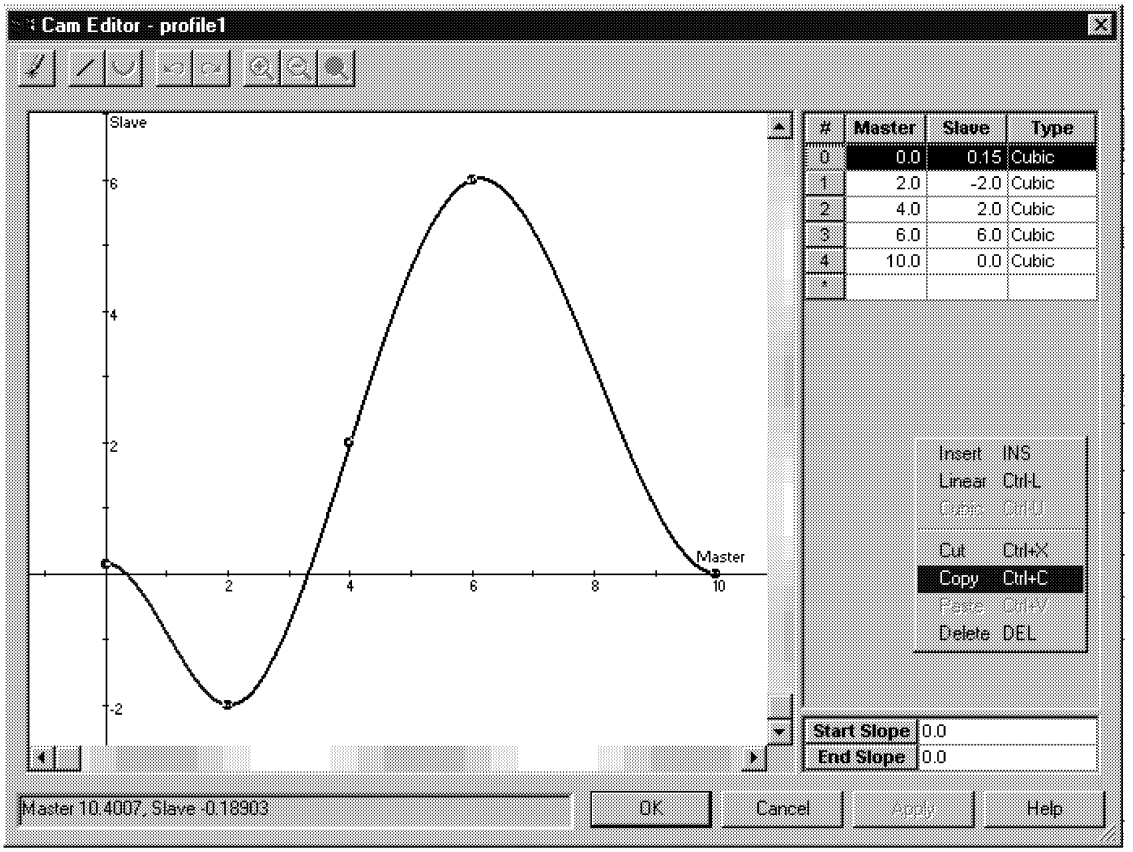


Figure 5 d: Cam Editor for MCCP instruction

>>Go to Rung 0 in the TCAMwithMCCP Routine. Open the MCCP instruction, CamProfile, […] box, next to the profile1[0] tag.

>>Select the first Master point 0 in the Cam Editor – profile 1 using a left mouse click.

>>Right mouse click and select Paste. The data was copied in the Excel Spreadsheet will now be pasted in this table. The actual Cam Profile will automatically be drawn.

Now, you can also edit the Cam Data profile 1 from the Program Tag editor of RSLogix 5000. Simply go to Program Tags, Main Program Scope. Open the [+] of the camdata tag. There are 5 points defined. Open the [+] of camdata[0]. Notice the Master time, Slave and curve type when you are in the "Monitor" mode. See Figure 5e. If you open camdata[1], camdata[2}, camdata[3] and camdata[4] these values should match the exact information you copied from the Excel Spreadsheet.

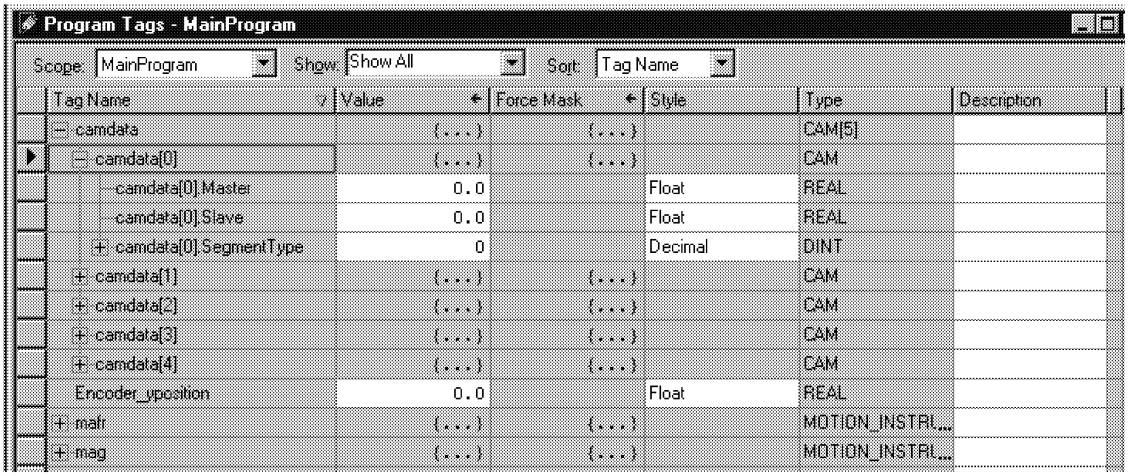


Figure 5e: Edit Cam Data from Program Tags

Monitoring CAM_PROFILE with the CAM editor

Even though the contents of a tag of type CAM_PROFILE were created using the MCCP call instead of the Graphical Cam editor, the editor can still be used when on-line to display the contents of the CAM PROFILE. To launch click on the dotted box next to the CAM field in the MCCP call or on the PROFILE1 tag in the MATC call. See Figure 5f

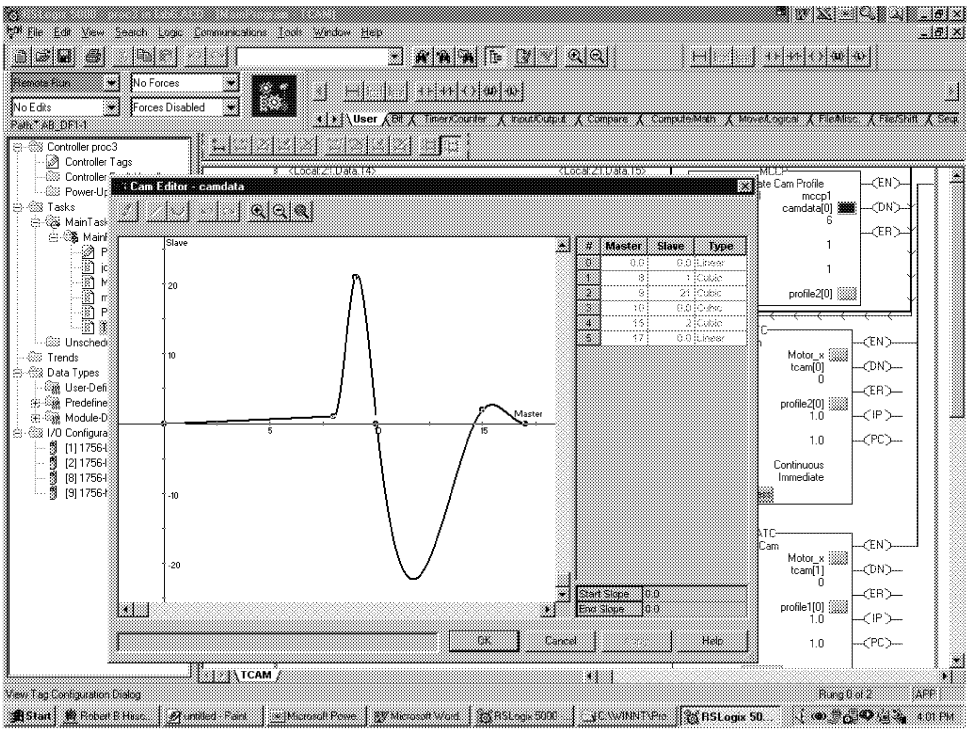


Figure 5f – Cam editor

Lab Exercise

1. Verify, Download and run the Proc1 program. Run (Note program has already been setup for proper path using ENET connected to processor in slot 1). If you weren't able to download the file, select the Communications tab in RSLogix 5000. Go to RSWho. Expand the Ethernet, TCP-1 tree, via the backplane and choose the L1 processor in slot 3. Check "Apply Current Path to the Project". Select Apply. Select Download.
2. Turn Switch DI12 to home the Motor_x Axis
3. Turn Switch DI14 on. Notice the motion on Motor_x. Repeat step 3 with DI15 on and note the difference. You can also open Trends to Run the chart to monitor Motor_xposition.
4. If a Motion Fault occurs, simply toggle DI12 switch and restart the lab.

Appendix A

How To Configure an ETHERNET Card

I. Setup Your PC

Under the folder Settings/Control Panel/Networks..
Go to the Protocols Tab
Find the TCP/IP Protocol Selection
Select TCP/IP and then select Properties
Specify an IP Address
- We used 131.130.130.129
Specify Bit Mask
We used 255.255.0.0
Select Apply, then OK

If you do not have a TCP/IP Protocol Selection, select ADD and try to find the file in location C:\386. Or if you have a disk, insert the disk to load this protocol. Select OK. If all else fails, call me or Lewis Anyanwu to discuss further.

After completing this exercise, you may need to reboot the computer.

II.Setup the ENET Card using CLGX Gateway Software

Open Control Logix Gateway Software. If you do not have the Gateway Software, go to www_tempest.cle.ab.com and download.
Select New File
1756-ENET configuration

Select the Port Configuration Tab
Specify an IP Address different than the PC, TCP/IP Address.
We used 131.130.130.130
Specify Subnet Mask
We used 255.255.0.0
Disable Bootp
Save to File Name. Call it as you wish. We chose ENETconfig
Go to File/Browse Network.
Connect the 1756-CP3 cable from your PC to a L1 processor that you want to communicate with.
Select in the Browse Network or RS Who the L1 processor that you will work with.
Hit the + button to expand that L1 processor.
Find the Backplane, expand the + button.
Find the ENET Card and highlite it. Select OK.
Go to File/Send File to Module.
Make sure your computer is connected to the ENET Card in CLGX with the provided ENET cable.
Open ENET config(or whatever you named it) and hit OK.
Exit.

You have now configured your ENET card using Control Logix Gateway.

III.Configure Drivers

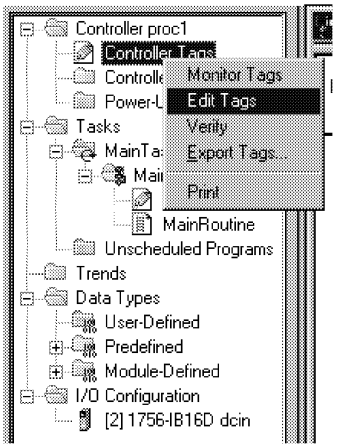
Go to RS Linx Communications.
Select Configure Drivers.
Select “Remote Devices via Linx or 1756-ENET Gateway”.
Add New
Choose a name for this Linx. We chose TCP-1 (this is default).
Select OK.
Under Driver Selection Tab
Server’s IP Address is same as the ENET card setup.
131.130.130.130
Under Configure Browser Tab
Network Number 131.130.130.130. Hit Add.
Select Apply
Select OK
Under Configured Drivers you should see, “TCP-1 to on 131.130.130.130 Running”.

Hit Close.
RS Who active should now show a TCP-1 Ethernet linx.
Hit the + to expand the RS linx and show that you can communicate to the back plane through the Ethernet Network.
Select the appropriate L1 processor or card that you wish to address.
Note , you can use this process to program and flash card firmware.

Appendix B

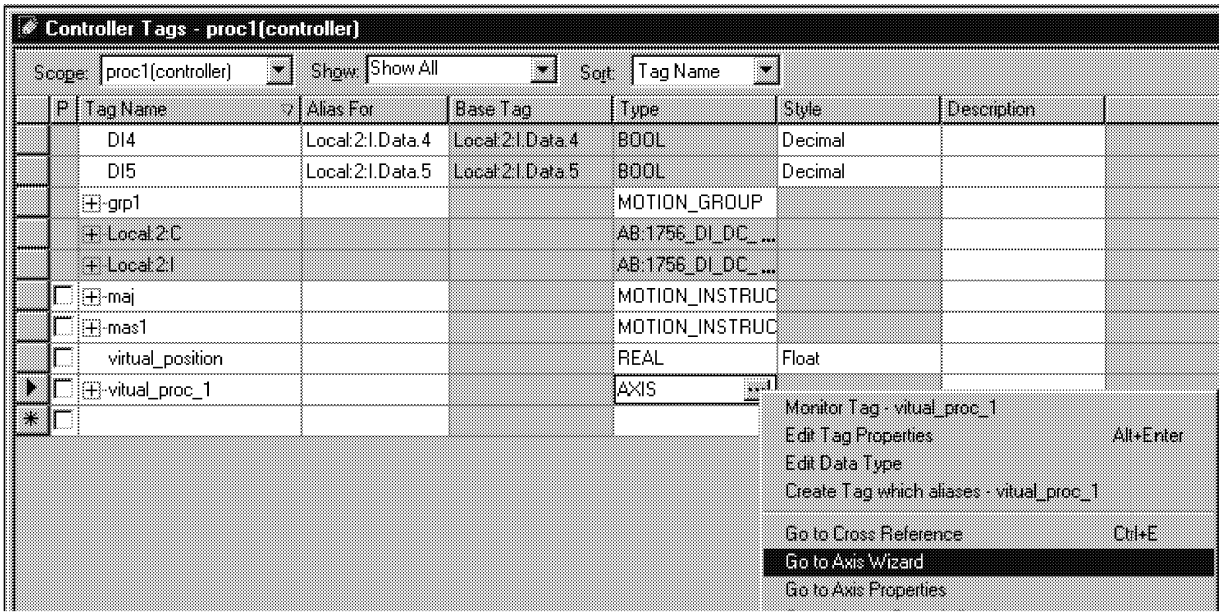
Create a Virtual Axis

>>Select Edit Controller Tags by right click of Controller Tags and select Edit Tags (assuming you have a project already opened)

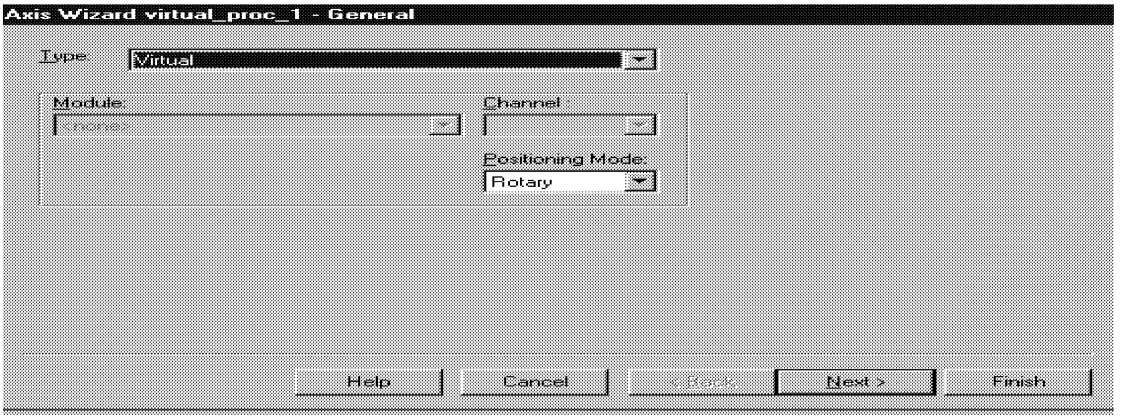


>>Type in tag name … In this case, for an example, we used the graphics of another program which used virtual_proc_1 as another virtual axis….Select data type of AXIS. If not done so already, select the box under “P” which will allow this axis to be produced on the Producer-Consumer Model Backplane (note this is so the axis can be consumed in later labs)

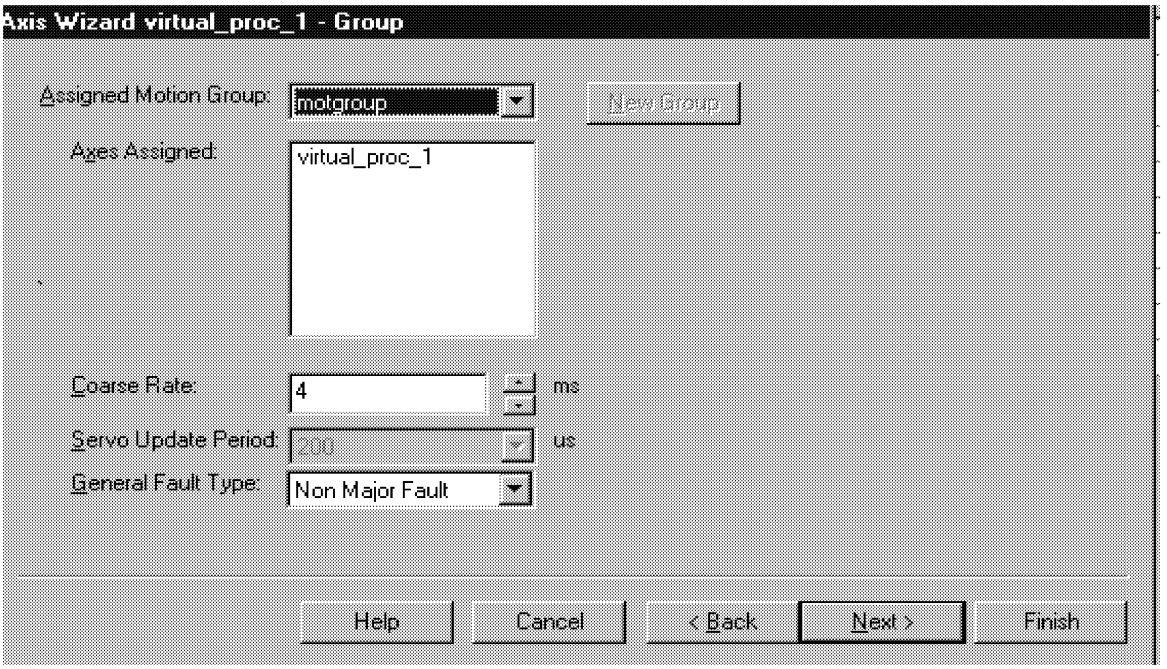
>>Place cursor under the Description data field, right click, and select Go To Axis Wizard



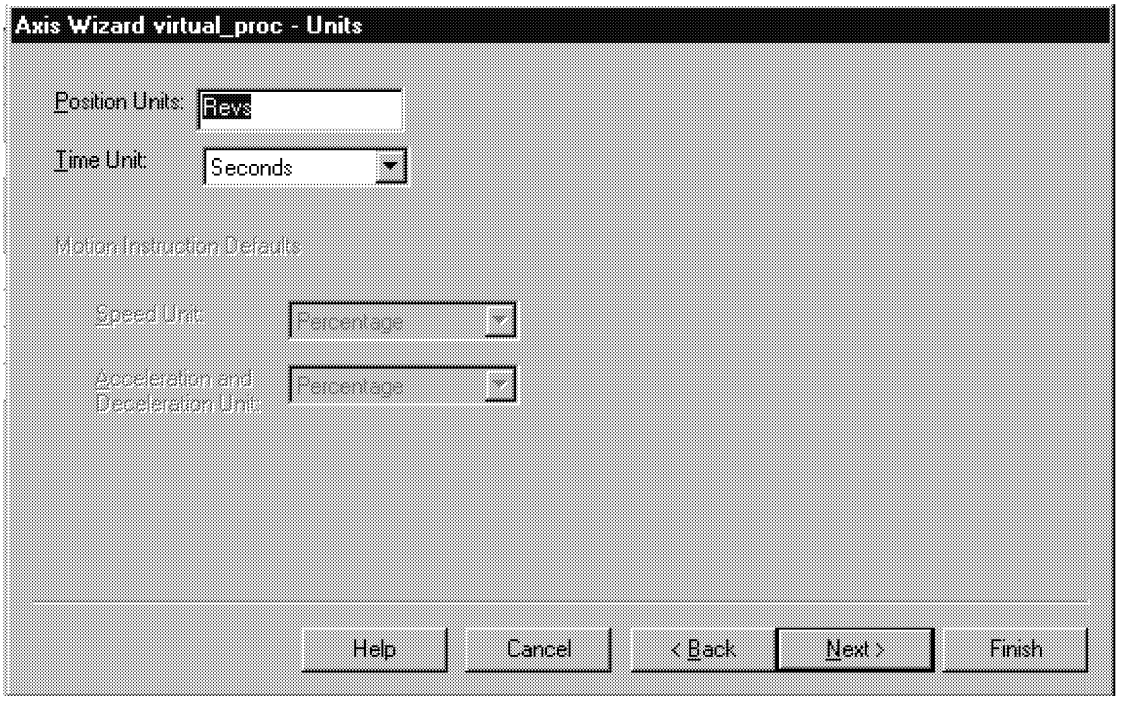
- Enter the following information in the axis wizard:



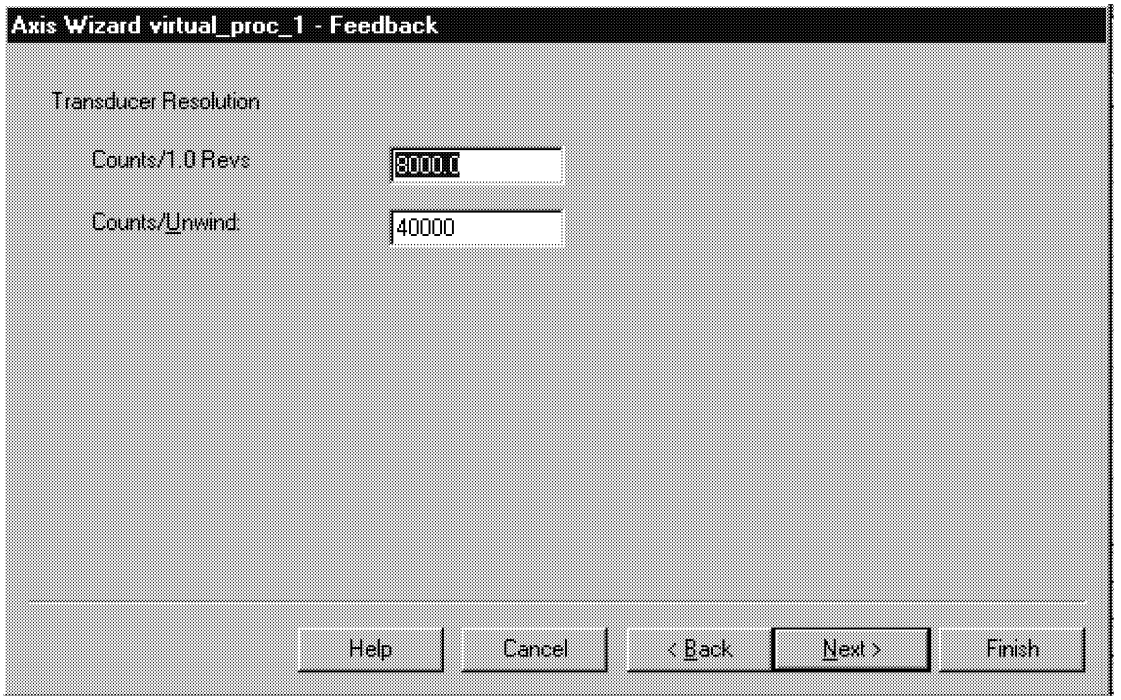
Select Next



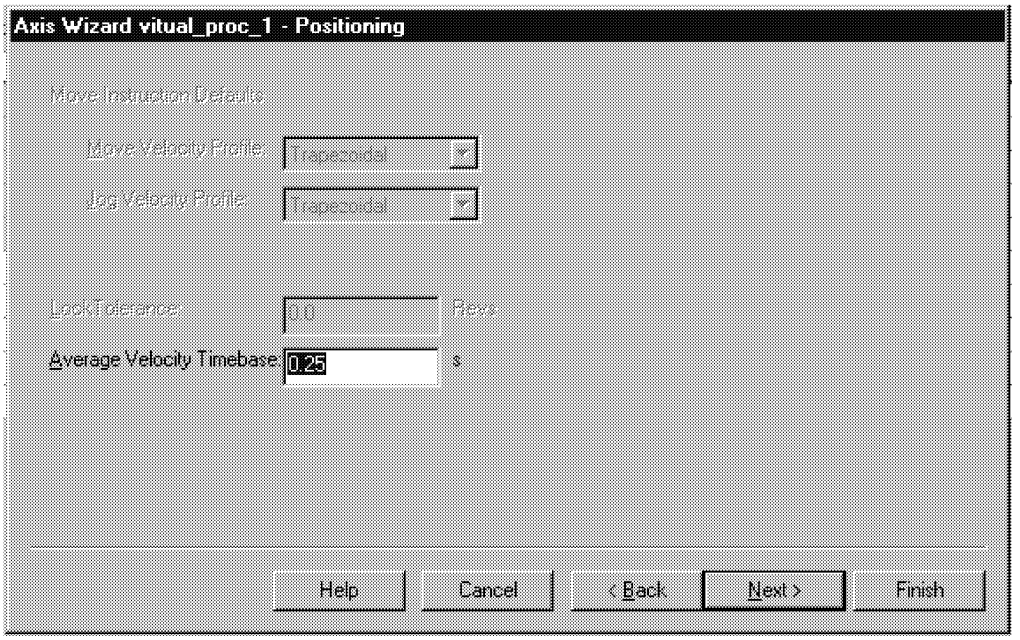
- Select Next



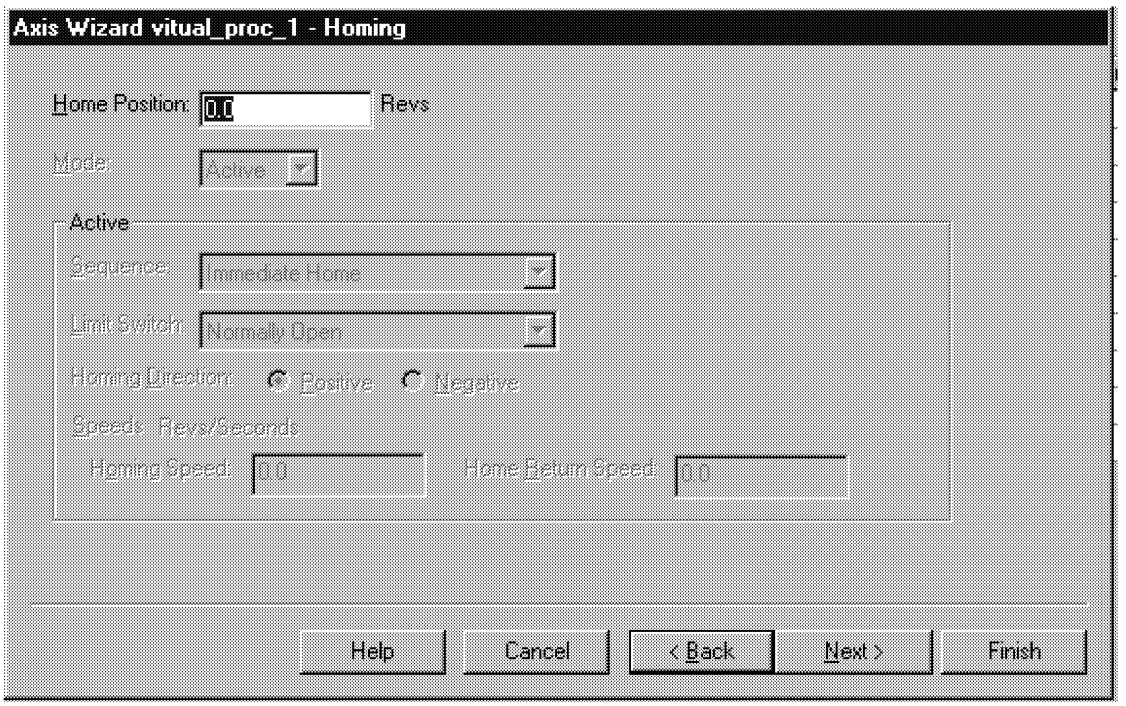
- Select Next



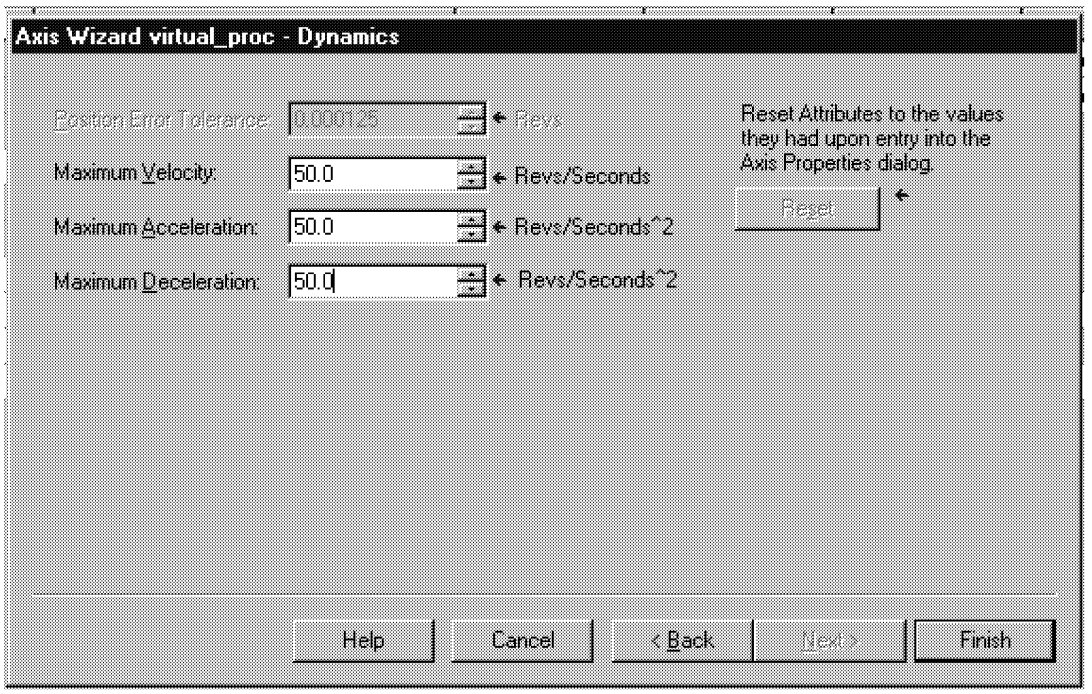
- Select Next



- Select Next



- Select next until the Dynamics dialog appears
- *Note: The following fields will be "grayed out" because they are not applicable to a virtual axis: Overtravels, Servo, Fault Action, Hookup, Tune & Gains*


Axis Wizard virtual_proc - Dynamics
Position Error Tolerance: 0.000125 ← Revs
Maximum Velocity: 50.0 ← Revs/Seconds
Maximum Acceleration: 50.0 ← Revs/Seconds^2
Maximum Deceleration: 50.0 ← Revs/Seconds^2
Reset Attributes to the values they had upon entry into the Axis Properties dialog.
Reset
Help
Cancel
< Back
Next >
Finish

- Select Finish

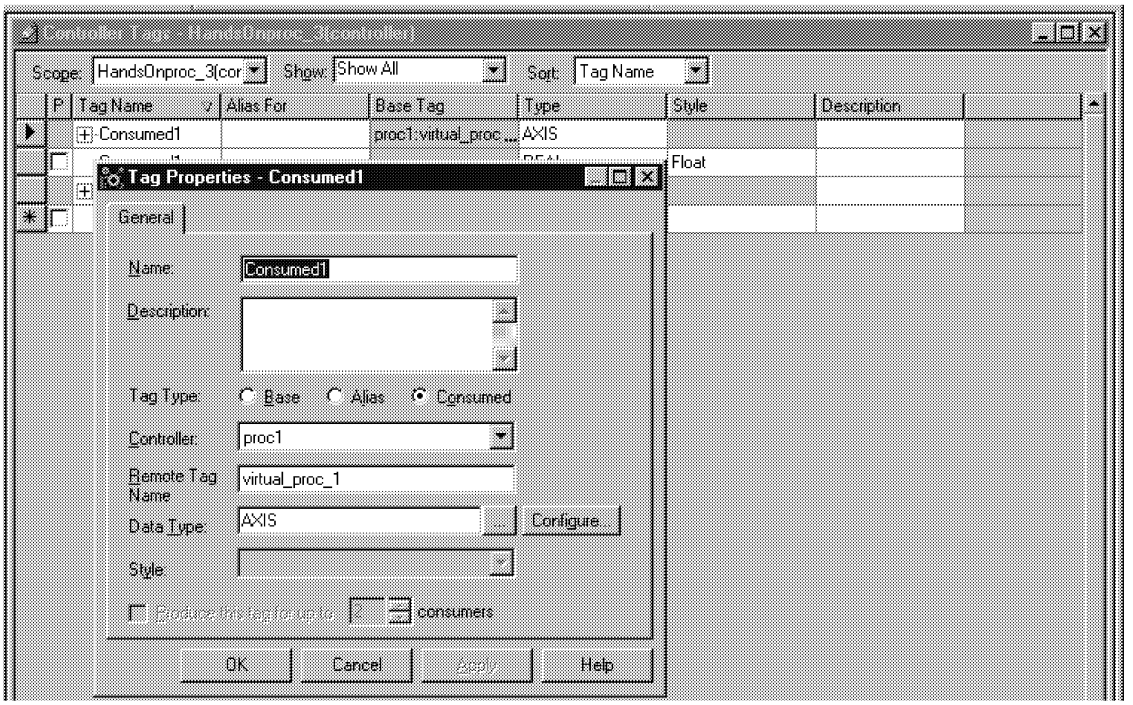
Appendix C

Create a Consumed Axis

>>You''ll need to define one of the 1756-L1 processors as an I/O Configuration with the Axis you are going to consume over the producer-consumer backplane.

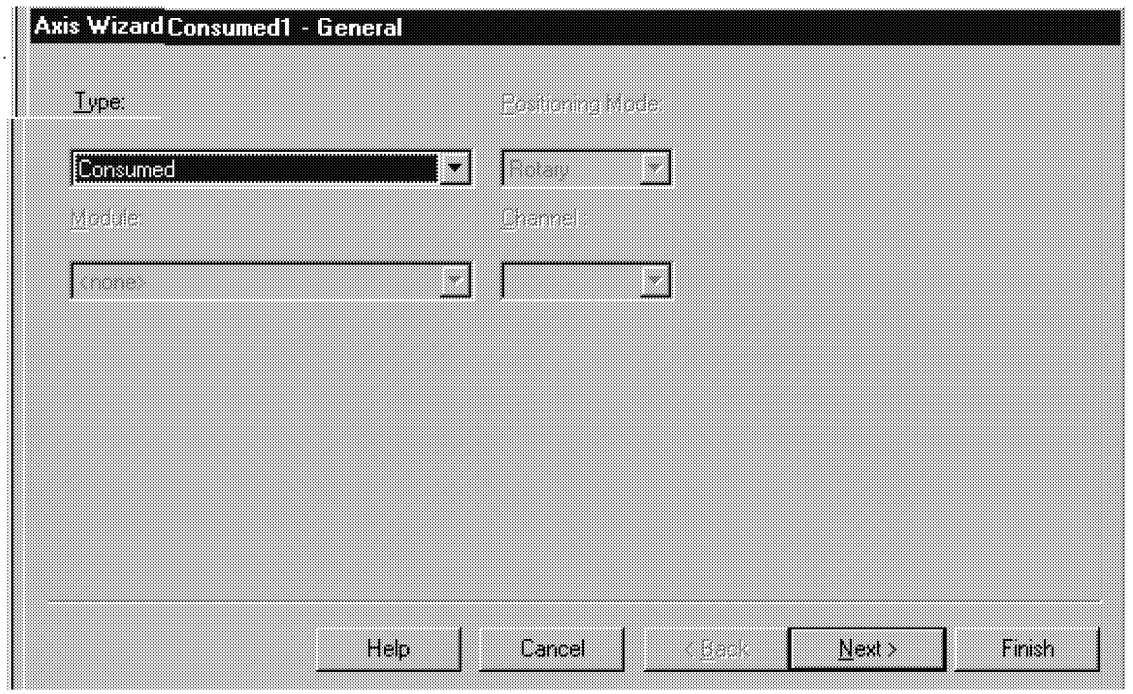
>>Select Edit Controller Tags by right click of Controller Tags and select Edit Tags

>>Name the Axis as Consumed1. Under Data Type, select this as an Axis. Under description, right-mouse click and Edit Tag Properties. Tag type should be consumed. Controller should be the processor you are consuming the tag from which in this case is proc_1. Remote tag name should be the axis you are consuming which in this case is virtual_proc_1.

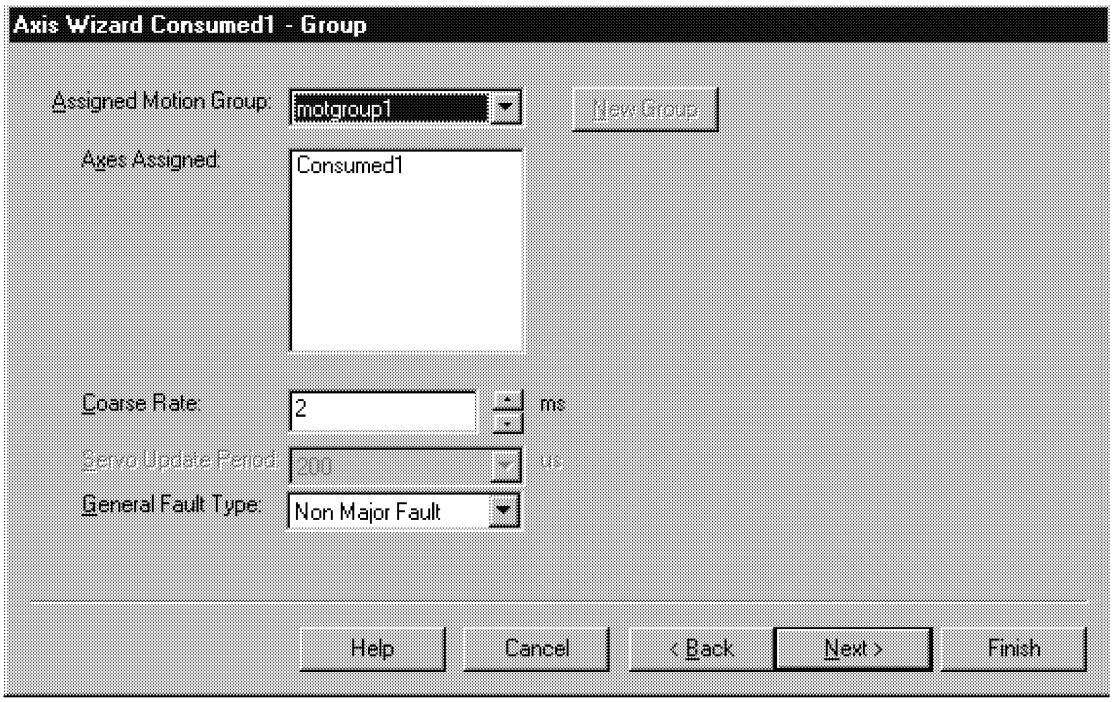


>>Select the Configure tab.

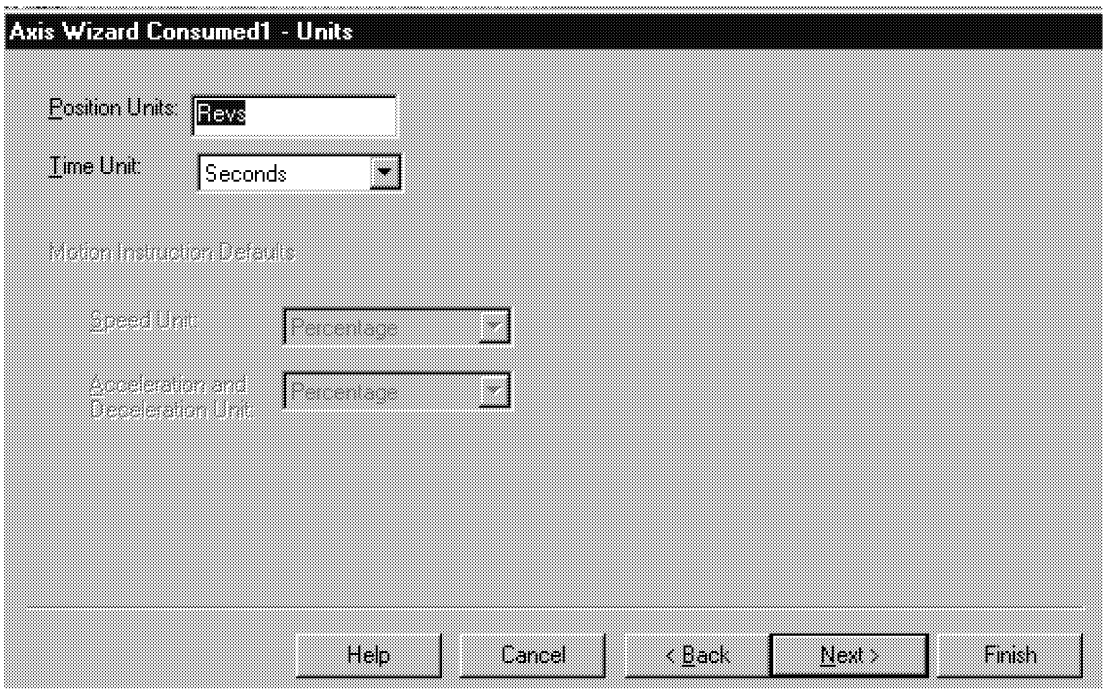
>>Enter in the following information:



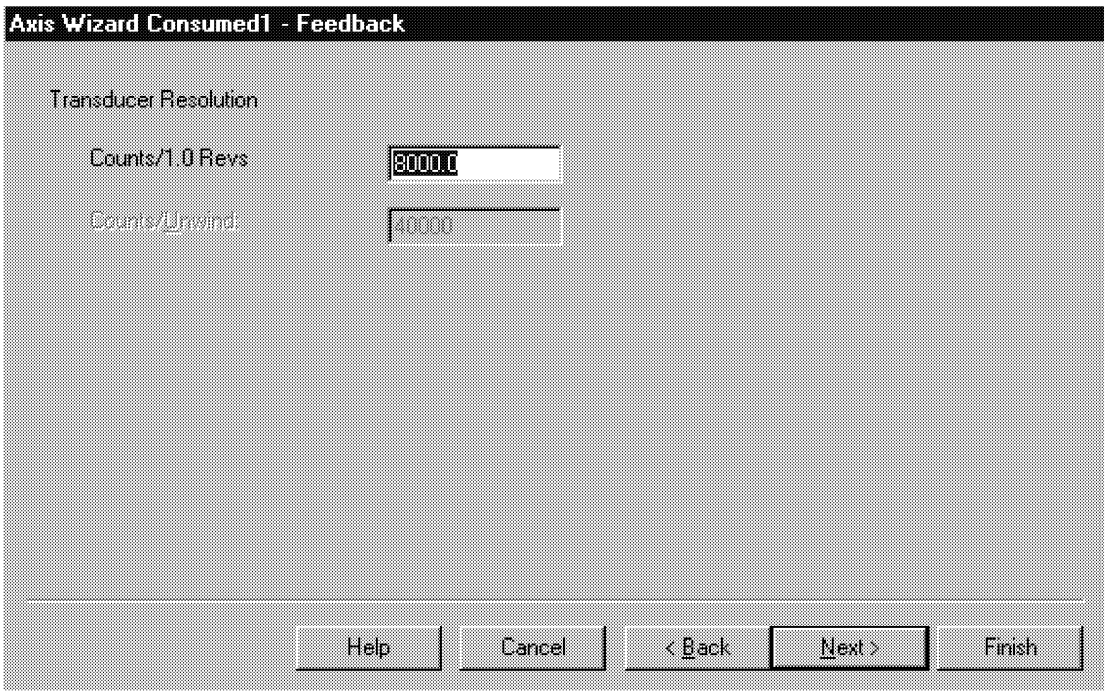
>>Select Next



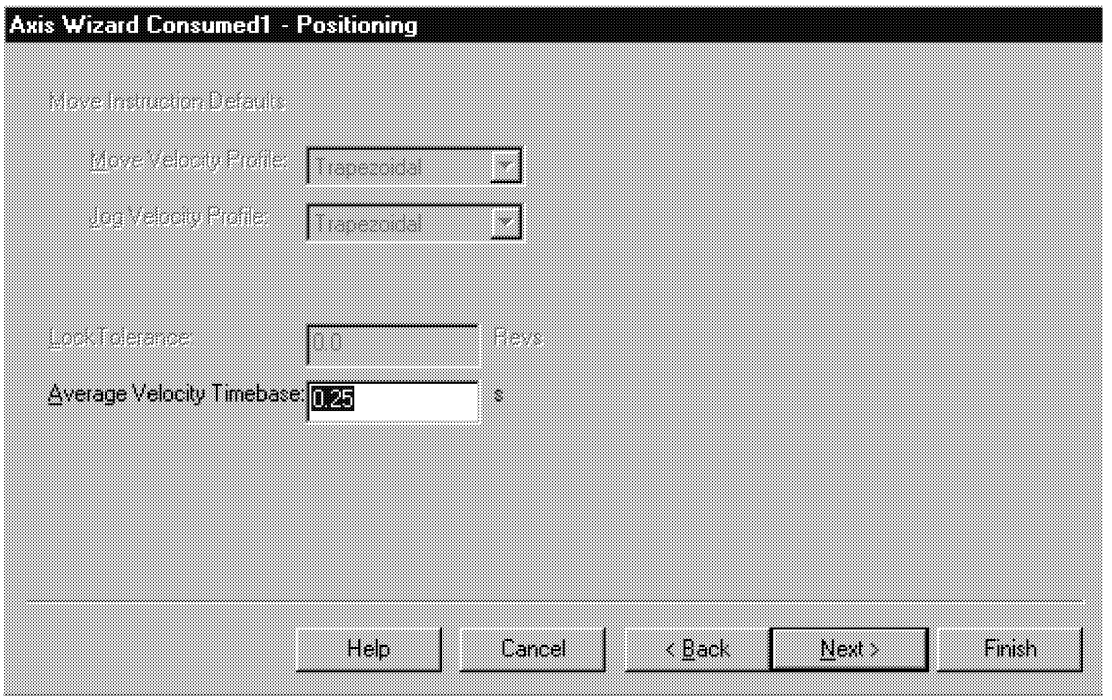
>>Select Next



>>Select Next



>>Select Next



>>Select Next

>>The next folders, Homing, Overtravels, Servo, Fault Action, Hookup, Tunes, Gains and Dynamics have no information about them since this information is contained in the Axis you are producing which is now being consumed. Select Finish to complete this Task.

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

Publication HOM_ADV Logix, January 2000